DinoBank Security Assessment

Team: [redacted]

November 23-25, 2019

Document Properties Page

Client	DinoBank
Title	[redacted] Penetration Test
Target	DinoBank IT Infrastructure & Systems
Version	1.0.0
Pentesters & Author	[redacted]
Approved by	Tom Dickson, Senior Information Security Officer

Version Control

Version	Date	Author	Description
1.0.0	24 November 19	[redacted]	Initial Version

Contact

Name	The [redacted] Pentesting Group
Address	445 SMB Way, Contoso, WA, USA
Phone	(123) 456-7890
Email	[redacted]@dinobank.us

1. Executive Summary **6**
1.1 Overall Severity Rating 7
1.2 Key Findings and Recommendations 7
1.3 Strategic Guidance 8
1.3.1 Short Term Recommendations 8
1.3.2 Long Term Recommendations 8
1.4 Scope 9
1.5 Rules of Engagement 9

2. Methodology and Terminology **11**
2.1 Approach 11
2.2 Risk Classifications 11
Critical 11
High 12
Medium 12
Low 12

3. Phases and Test Timeline **13**
3.1 Phase 1 - Open Source Intelligence Gathering 13
3.2 Phase 2 - Network Reconnaissance 13
3.3 Phase 3 - Weaponization and Delivery 16
3.4 Phase 4 - Privilege Escalation and Lateral Movement 16
3.5 Phase 5 - Actions on Objectives 16

4. Governance, Risk, and Compliance Audit Analysis **17**
4.1 Security Governance 18
4.2 Password Policy 18
4.3 Banking Core Applications 19
4.4 Outdated Software 20
4.5 Management 21
4.6 Audit Coverage 21

5. Findings **23**
5.1 Critical Severity Findings 24
5.1.1 Postgres Metasploit Module (SID-101) 24
5.1.2 Unauthenticated Money Transfer (SID-102) 26
5.2 High Severity Findings 28

3
5.2.1 Credentials on Internal Wiki (SID-201) 28
5.2.2 Password Reuse (SID-202) 30
5.2.3 Weak Passwords (SID-203) 30
5.2.4 FileZilla Anonymous FTP (SID-204) 31
5.2.5 Passwords in PowerShell Script Block Logging (SID-205) 33
5.2.6 Passwords Stored in Plaintext File (SID-206) 34
5.2.7 Data in SYSVOL Share (SID-206) 35
5.2.8 Unnecessarily Exposed Services (SID-207) 36
5.3 Medium Severity Findings 38
5.3.1 MediaWiki Edit Permissions (SID-301) 38
5.3.2 Anonymous SID/Name Translation (SID-302) 39
5.3.3 Malware Discovered (SID-303) 40
5.3.4 Weak and Unenforced Password Policies (SID-304) 41
5.3.5 Firewall Disabled (SID-305) 42
5.3.6 Malicious Scheduled Tasks (SID-306) 43
5.3.7 XSS on Trading Site (SID-307) 45
5.4 Low Severity Findings 47
5.4.1 Media Wiki Arbitrary Account Creation (SID-401) 47
5.4.2 No logging on Filezilla FTP (SID-402) 48
5.4.3 FileZilla Bounce Attack (SID-403) 49
5.4.4 Improper TLS Implementation on Web Servers (SID-404) 51
5.4.5 IVR Server Exposure (SID-405) 52
5.4.6 Anonymous Share Enumeration (SID-406) 53
5.4.7 Ether web API source code disclosure (SID-407) 54
5.4.8 Usage of SMBv1 (SID-408) 55
5.4.8 Outdated Apache Server Version (SID-409) 55

6. Non-Findings 57
6.1 Access Control 57
6.1.1 Hardened SSH Server (NF-101) 57
6.1.2 Restrict RPC Null Session (NF-102) 57
6.1.3 Lack of linux privilege escalation (NF-103) 57
6.2 Input Validation 57
6.2.1 SQL injection input sanitization (NF-201) 58
6.2.2 Lack of XSS on Wiki (NF-202) 58
6.3 Network Access 58

6.3.1 MSSQL Listening on Loopback Adapter (NF-301) 58
6.4 Credential Management 58
6.4.1 Password hashes resistant to cracking (NF-401) 59
6.4.2 SSH using private key authentication (NF-402) 59
6.5 Data Security 59
6.5.1 MediaWiki was logged (NF-501) 59
6.5.2 Centralized Windows logging (NF-502) 59
6.6 Software Updates 59
6.6.1 Updated Linux Kernel (NF-601) 60
6.6.2 Updated Windows Version (NF-602) 60

7. Comparison to Previous Engagement 61
7.1 Overall Summary 61
7.2 Key Improvements Identified 61
7.2.1 Unauthenticated Account Creation 61
7.2.2 Croissant Bakery Removed 61
7.2.3 SQL Injection, Account Number 62
7.2.4 XSS to steal PHPSESSID cookies 62
7.2.5 Password Policy 62
7.2.6 Static API Keys 62
7.2.7 Information Leak 63
7.3 Previous Key Findings Not Yet Remediated 63
7.3.1 PostgreSQL 63
7.3.2 FileZilla Anonymous FTP 63
7.3.3 Null SMB Sessions on Branch Domain Controller 63
7.3.4 FTP Bounce Attack 63
7.3.5 Arbitrary Money Transfer 63
7.3.6 Exposed Domain Controller 64
7.3.7 Exposed PostgreSQL database 64
7.3.8 Exposed Remote Access Services 64
7.3.9 Plaintext Credentials and PII (PF-401) 64
7.3.10 HTTP over HTTPS (PF-501) 64
7.3.12 Outdated Apache Web Server (PF-601) 64
7.3.13 Outdated SMB (PF-602) 65

8. Special Assessment Areas 65

8.1 CroissantCoin Blockchain-based banking solution 65
8.2 Interactive Voice Response (IVR) System 66
8.3 Automated Teller Machine (ATM) 67
8.4 Active Directory Environment 67
Stale Objects (Medium) 68
Domain Computer Registration Process 68
SMBv1 is Enabled (Low) 69
Privileged Accounts (Critical) 69
More than 95% of AD Administrative Accounts are inactive 69
Presence of Admin accounts which lack the flag "this account is sensitive and cannot be delegated" 70
Ensure that dangerous settings & privileges are not granted to everyone by GPO 70
Trusts (Non-Finding) 71
Anomalies (High) 71
All local Administrator accounts have the same password 71
Weak account policy requirements 72

9. Conclusion **73**

1. Executive Summary

[redacted] was asked to return for a second penetration test of the DinoBank network. This engagement is in response to a Memorandum of Understanding issued to DinoBank by the Gotham Department of Banking Office of the Comptroller of Currency. The primary intent of the penetration test was to audit DinoBank networks according to the guidelines set forth by this governing body. An important secondary purpose of the penetration test was to assess how well DinoBank responded to the previous engagement. This engagement was to occur during business hours without interfering with DinoBank operations, and was scoped to include full access to a network of five subnets, an interactive voice response unit, and an Automated Teller Machine.

During the test, we were able to identify numerous services that contained sensitive customer and employee data, which could be accessed with well-known attack techniques. This enabled the team to obtain sensitive customer data such as social security numbers, emails, and passwords. We also found cleartext passwords for the Active Directory, this allowed us to access significant portions of the Windows infrastructure, which could be used to read arbitrary emails or to modify employee data. Additionally, we identified points in the DinoBank web application that allowed us to transfer funds between accounts without authentication. Finally, we found a critical flaw in the ATM that allowed us to authenticate ATM cards using any four or more digit pin.

In regards to the MOU issues by the Gotham Department of Banking Office of the Comptroller of Currency, DinoBank was found satisfactory or strong is a couple of categories, but unsatisfactory -- well below average -- in a number of important criteria, including security governance, core banking applications, and management.

It was found that DinoBank did not adequately respond to our previous engagement. None of the critical vulnerabilities, which allowed for the exfiltration of sensitive customer and employee data, were fixed.

There were a few important non-findings. DinoBank did a great job logging and auditing across their systems. Their operating systems and third party services were generally up to date and secure. We were unable to escalate privileges on the Linux systems, and we failed to get access to others.

1.1 Overall Severity Rating

Critical

Numerous instances of extremely sensitive customer and employee data were identified and accessed by the team. Furthermore, malicious transfer of money between accounts was found to be possible. This presents a large issues to customers and employees by compromising their privacy and eroding trust in DinoBank. Finally, we were able to obtain access to large portions of the DinoBank network, which would allow a malicious attacker to persist across the network and attempt to further spread to core banking services.

1.2 Key Findings and Recommendations

Area	**Observation & Impact**	**Recommendation**
Access Control	Unauthorized access to FTP server and PostgreSQL server containing sensitive employee and customer data.	Move PostgreSQL server to private database server and move the FTP server to a private segment of the network.
Network Segmentation	Domain controllers, remote access services such as RDP and SSH, and internal websites such as the MediaWiki were publicly exposed on the network.	Segment the network to prevent public access to sensitive ports, especially on the Domain Controller.
Credential Management	Plaintext passwords were stored throughout the environment. Weak passwords were reused between important accounts.	Always store sensitive data securely by hashing it. Password policies should require long, complicated passwords than cannot be reused.

1.3 Strategic Guidance

The following section covers short term and long term guidance on how to improve the security of your network. Short term recommendations are intended to give a list of actions that can quickly improve the security of the systems, while long term recommendations are intended to be extended behaviours and policies that must be implemented and maintained.

1.3.1 Short Term Recommendations

Our two most important short term recommendations are to change local Administrator passwords in Windows workstations and to securely handle sensitive data such as passwords and SSNs across DinoBank systems. While this might not be an easy change to implement, it is vital to the long term health of DinoBank.

Our other short term recommendations are as follows:

- Change local Administrator passwords on Windows workstations to longer, more complicated passwords that are not reused
- Securely handle PII such as SSNs and passwords in databases and other services using hashing and encryption algorithms
- Disable unauthenticated access to services such as FTP and PostgreSQL
- Patch arbitrary money transfer in customer portal on 10.0.2.101
- Disable null SMB session on the Gotham Domain Controller
- Disable SMBv1 across the Windows network

1.3.2 Long Term Recommendations

Across DinoBank systems, there was poor network segmentation and externally exposed services. The network needs to be properly segmented to make lateral access more difficult, and most services should be hidden behind an external firewall unless customers need them. This will hamper attackers from being able to enter and traverse the network without proper authentication.

In addition, DinoBank's password policies were below average. The local Administrator accounts were very weak, a and there were inadequate password policies. Robust password policies should be developed and implemented across both Windows and Linux systems, for all machine and service accounts.

Finally, there were many insecure software practices exhibited across DinoBank, ranging from poor coding practices to leaving plaintext credentials in files. DinoBank should invest in training for its employees regarding secure software practices to prevent these issues from repeating or worsening.

1.4 Scope

The scope of this assessment was limited to the following five subnets: 10.0.1.0/24, 10.0.2.0/24, 10.0.10.0/24, 10.0.11.0/24, and 10.0.12.0/24. This means all services running on ports in this IP range were subject to testing for the duration of the assessment. These services included DinoBank's proprietary blockchain-based banking solution, several custom web applications, DinoBank's core and web banking services, and their Microsoft Active Directory Environment. In addition, we were asked to test an ATM and an IVR phone system with the caveat that we could not physically test the ATM and we could only test the IVR unit by manually dialing the number on speaker phone.

1.5 Rules of Engagement

For a period of twelve hours split between two days, testers were allowed to take any action for reconnaissance, privilege escalation, or command execution that would improve the quality of the security provided those actions not hamper DinoBank's daily operations. Testers were able to utilize any gained information to further access in order to better simulate a malicious attack, this includes potentially sensitive information such as credentials. In terms of modification ability, testers were permitted to download external software, tools, or files onto company systems if this action was considered necessary for a more complete test. However, no data was permitted to leave the controlled DinoBank environment. If critical vulnerabilities were found, they were to be reported immediately so that remediation measures could be taken.

As DinoBank was attempting to continue daily operations for the duration of the test, a priority of the engagement was to not impact any services critical to DinoBank's daily functionality. In addition teams were to limit load on network segments caused by testing to prevent degradation of performance speed for the customers. Methods

involving extensive social engineering or physical access to DinoBank systems were not in scope.

If a team had any pertinent and reasonable questions, they were to either contact DinoBank's Information Security Officer or to email a provided support address.

2. Methodology and Terminology

2.1 Approach

Before the team had network access, we conducted open source intelligence (OSINT) gathering on the DinoBank team members and services to gain an initial understanding of the services offered by DinoBank as well as potentially useful information for password spraying. We then began the assessment with reconnaissance by conducting standard network scans to discover public facing services. This gave us an initial understanding of the network we were working with. From this point, the team began applying methods for potential vulnerability exploitation on the externally facing services.

As vulnerabilities were identified and in particular as code execution was gained on a system, the team performed additional reconnaissance to locate sensitive data accessible by low privilege users and gain greater knowledge of network services. We then attempted local privilege escalation from basic shell access. Concurrently, members of the team worked to assess the security of the IVR system and ATM.

2.2 Risk Classifications

Below are explanations of the risk classification system utilized by this team in describing both present and absent security risks. All risks have been assigned a classification to assist the client in prioritizing vulnerabilities based on impact to the company. Any generalities, such as “low” or “medium” have been assigned by this team for better depiction of a risk.

Critical

A critical vulnerability would generally allow an attacker to take control of a system or provide inexcusable access to sensitive personal data. These are of imminent concern to DinoBank, and include the following:

- Unauthenticated Remote Code Execution (RCE) with low to medium ease of exploitation on a system that has access to critical processes or PII
- Unauthenticated disclosure of highly sensitive Personally Identifiable Information (PII)

High

High risk vulnerabilities are characterized as those that allow access to machines or important company information. When combined, high risk vulnerabilities can often escalate to critical ones. The following are standard risks that fall into this category:

- Authenticated RCE
- Unauthenticated RCE that is difficult to execute
- Privilege escalation vulnerabilities requiring minimal system knowledge
- Authenticated disclosure of high impact PII
- Unauthenticated disclosure of low to medium impact PII
- Disruption of the Integrity of Existing Data

Medium

Medium risk vulnerabilities are characterized as those usable only by attackers highly knowledgeable in system information, ones that provide system knowledge, or low-risk service misconfigurations. The following are broad categories for this risk level:

- Network misconfiguration for both hosted services and system versions
- Privilege escalation vulnerabilities requiring extensive system knowledge
- Authenticated disclosure of low to medium impact PII

Low

Low risk vulnerabilities are primarily those that unnecessarily expose non-sensitive information about the network to attackers.

- Information disclosure about service and system versions without direct access
- Services with high potential for future exploitability
- Web servers lacking proper TLS implementation
- IVR issues of note

3. Phases and Test Timeline

3.1 Phase 1 - Open Source Intelligence Gathering

The engagement began with performing open-source intelligence (OSINT) on DinoBank. We identified accounts belonging to employees of the company as well as software that seemed of particular interest to the company. We found a number of employees via Linkedin, Twitter, Reddit, and Github. This resulted in us creating a list of potential usernames for future password spraying attempts.

The company GitHub repository also proved to be interesting. Although the majority of the forked repositories did not provide much insight, we did find that one user, DinoDanOliver, had checked in both a private and public SSH RSA key. Finally, we found a proprietary bitcoin and program called “croissant coin”.

3.2 Phase 2 - Network Reconnaissance

For the next phase of our assessment we scanned the five provided subnets. The results of our initial scans are presented below where we display the IP, Hostname, Role, and Key Services identified during the scan.

10.0.1.0/24 - Internal Network

IP Address	Hostname	Role	Key Services
10.0.1.10	CORP-DC-01	Domain Controller	DNS, Active Directory Domain Services, SMB, RDP
10.0.1.11	CORP-DFS-01	DFS	RPC, SMB, RDP
10.0.1.12	CORP-WSUS-01	FileZilla FTP 0.9.60	FTP, HTTP, RPC, SMB, RDP
10.0.1.20	CORP-EXCH-01	Microsoft Exchange	Microsoft Exchange Server,

		15.1.1779.2	SMB, RDP
10.0.1.31	CORP-WEB-01	MediaWiki 1.32	HTTP, RPC, MySQL, SMB, RDP
10.0.1.33	CORP-WEB-03	Notary service	SSH, HTTP, HTTPS
10.0.1.50	WAREHOUSE	MSSQL	RPC, SMB, RDP
10.0.1.115	JGAY-WS	Workstation	RDP
10.0.1.116	TDICKSON-WS	Workstation	RDP
10.0.1.250	COINS-01	Cryptocurrency Trade	SSH, HTTP, HTTPS

10.0.2.0/24 - CORE Network

IP Address	Hostname	Role	Key Services
10.0.2.100	CORE-01	PostgreSQL Server	SSH, HTTP/S, PostgreSQL
10.0.2.101	BANKWEB-01	OnlineBanking	SSH, HTTPS
10.0.2.102	IVR-01	IVR Management	SSH, Asterisk Management.
10.0.2.103	REPORTS-01	QueryTree	SSH, HTTP
10.0.2.113	HEADS-01	HTTP server	SSH, HTTP/S
10.0.2.115	TAILS-01	Ether search	SSH, HTTP/S, Nodejs
10.0.2.200	WIRES-01	SSH server	SSH

10.0.10.0/24 - Gotham Branch Network

IP Address	Hostname	Role	Key Services

10.0.10.100	GOTHAM-DC	Domain Controller	DNS, Active Directory Domain Services, SMB, RDP
10.0.10.201	GOTHAM-TLR-01	Teller	RPC, SMB, RDP
10.0.10.202	GOTHAM-TLR-02	Teller	RPC, SMB, RDP
10.0.10.203	GOTHAM-TLR-03	Teller	RPC, SMB, RDP
10.0.10.208	GOTHAM-WK-01	Workstation	RPC, SMB, RDP
10.0.10.209	GOTHAM-WK-02	Workstation	RPC, SMB, RDP

10.0.11.0/24 - Metropolis Branch Network

IP Address	Hostname	Role	Key Services
10.0.11.100	METRO-DC	Domain Controller	DNS, Active Directory Domain Services, SMB, RDP
10.0.11.201	METRO-TLR-01	Teller	RPC, SMB, RDP
10.0.11.202	METRO-TLR-02	Teller	RPC, SMB, RDP
10.0.11.208	METRO-WK-01	Workstation	RPC, SMB, RDP

10.0.12.0/24 - Springfield Branch Network

IP Address	Hostname	Role	Key Services
10.0.12.100	SPRING-DC	Domain Controller	DNS, Active Directory Domain Services, SMB, RDP
10.0.12.201	SPRING-TLR-01	Teller	RPC, SMB, RDP
10.0.12.208	SPRING-WK-01	Workstation	RPC, SMB, RDP

3.3 Phase 3 - Weaponization and Delivery

Once we were able to identify targets, we began the process of weaponization and delivery. We targeted web servers, the Postgres Database server, the FTP server, and the MediaWiki server. On the FTP server, we discovered that FTP bounce and anonymous access was enabled. We also found encoded credentials for the Domain Administrator account. We identified that the PostgreSQL service on 10.0.2.100 was vulnerable to null sessions, which we exploited using Metasploit. The MediaWiki server contained posts referencing cleartext credentials for Windows workstations, which we were able to modify slightly and use to gain access to Windows machines across the network.

3.4 Phase 4 - Privilege Escalation and Lateral Movement

After gaining preliminary access to the aforementioned servers, we focused our attention on privilege escalation. Although we were able to gain a shell and run commands on the Postgres server, it was unprivileged and fairly limited in capabilities. The credentials obtained through the MediaWiki server gave us access to local Administrator accounts, and we used these to connect to workstations and servers throughout the DinoBank network. These administrative credentials eventually enabled us to connect to the Domain Controller and obtain fully Domain Admin rights to the Windows Active Directory environment.

3.5 Phase 5 - Actions on Objectives

After spreading throughout DinoBank's systems, we focused on issues of interest to potential attackers and identifying other configuration issues. For example, we used our Domain Admin access to connect to the Microsoft Exchange server which gave us full control over any DinoBank email address. We also obtained access to file shares and database servers that are of interest to attackers, one of which that contained data on all DinoBank customers. Demonstrating access to customer PII and similar control of the DinoBank network helps to illustrate what real attackers might be after and audit DinoBank's adherence to their regulations and compliance requirements.

4. Governance, Risk, and Compliance Audit Analysis

The main motivator for this engagement is that DinoBank is currently being investigated by the Gotham Department of Banking Office of the Comptroller to discover if DinoBank is properly protecting their customers' PII. In order to help DinoBank make changes for this investigation, we have provided our own assessment regarding the following categories:

1. Lack of security governance
2. Weak passwords
3. Banking core weakness
4. Outdated software
5. Poor management
6. Insufficient audit coverage

A primary focus of regulatory guidance is to ensure that financial institutions have created an information security program that compiles with the Gramm-Leach-Bliley Act (GBLA) and is based on an assessment of risk. Accordingly, we have tailored this section to follow the IT Audit rating system of *Strong*, *Satisfactory, Satisfactory with Recommendations, Needs Improvement*, and *Unsatisfactory* and aligned our analysis to guidelines posted in the Federal Financial Institutions Examination Council (FFIEC) Information Technology Handbook.

Rating	Definition
Strong	Well above average performance
Satisfactory	Above average performance; no significant exceptions, or 1 or 2 minor exceptions
Satisfactory with Recommendations	Average performance; several minor exceptions or a single significant exception
Needs Improvement	Below average performance; several major exceptions

Unsatisfactory	Poor performance; violations of multiple high risk exceptions

4.1 Security Governance

Needs Improvement

Security governance refers to the systems by which an organization directs and controls IT security. It is different from management because it is more concerned with who is authorized to make decisions rather than how those decisions mitigate risk. Organizations with effective security governance have clear processes of communication and pathways by which decisions are made. While DinoBank did not have any high risk violations of security governance, there were many examples of poor governance.

During the engagement, on multiple occasions DinoBank employees approached [redacted] with requests to investigate other employees, asking for early information on the report, or to try and request special reports already sent to DinoBank management. It was clear that some of the employees were unsatisfied with the way information was disseminated from DinoBank executives and did not trust DinoBank leadership to properly decide which employees needed which information.

Furthermore, it was clear that some employees did not trust leadership to properly investigate suspicious behavior. There was doubt regarding the ability of executives to judge inside threats and properly respond, and as a result they tried to go around the chain of command and use the penetration test to influence DinoBank's decisions.

4.2 Password Policy

Unsatisfactory

Generally, DinoBank passwords ranged from medium strength to high strength. However, there are several severe exceptions with weak passwords that are used extensively throughout the network and with high permission accounts.

Where found, employee passwords in Linux systems were random, alphanumeric strings with special characters. These would not be easy to guess, but the short ones would not be terribly difficult to guess either.

Passwords to services such as the wiki, Reportasaurus, and core banking apps were 10+ characters, contained a special character, and sometimes contained a number. These were generally based on one long dictionary word or two short dictionary words, often relating to dinosaurs. These are also medium strength passwords.

Windows user accounts are also of at least medium strength. We obtained a hash dump and spent around six hours trying to crack them without success. However, we did not have access to powerful hardware or a dinosaur themed wordlist that may have made them much quicker to crack.

However, many Administrator accounts on Windows workstations used a very simple easy to guess password. It would likely be in the top 20 used passwords. There is also evidence that in the past, some normal AD user accounts used a simple variation on "DinoBank" with a single special character and no numbers. This password would likely be easily generated with a rule-based word list. Because this was such a simple passwords that was used so often and attached to high privilege accounts, we cannot give a good score for this category despite the medium strength passwords used elsewhere.

Related SIDs: SID-201, SID-202, SID-203, SID-206, SID-304

4.3 Banking Core Applications

Unsatisfactory

The CORE-01 server had a PostgreSQL database exposed with default credentials. This allowed [redacted] to gain remote code execution onto CORE-01 and view the database contents of all of the employees and customers, which allows attackers to steal money from DinoBank customers.

The BANKWEB-01 server was serving the online banking web application, where customers could login to view their bank account information and balance, as well as transfer money from one account to another through their bank account number. However, users could visit the balance transfer page without authentication to transfer money out of arbitrary bank accounts into their own accounts. Furthermore, attackers

could also specify their bank account number as both the sender and the receiver, allowing them to arbitrarily gain money through the transfer at the expense of Dinobank.

The rest of the servers in the Banking core are sufficiently updated and secure, as the operating systems are fully updated, and the web applications are fully updated or custom made. However, the aforementioned vulnerabilities will directly affect customer's trust towards Dinobank, so we cannot give a good score to this category.

Related SIDs: SID-101, SID-102, SID-207, SID-404

4.4 Outdated Software

Satisfactory

Across DinoBank networks, operating systems are fully updated and secure. All Windows machines were Windows Server 2016 Datacenter 14393, which is an LTSB. This means they are fully supported and secure. They are not vulnerable to any publicly known Windows exploits. Similarly, all Linux kernel versions that we were able to confirm were Linux 5.0.0, which is only a few months old. The only exception is that the Windows machines were configured to use SMBv1, which has been outdated and insecure for many years.

Most third party software that we can verify is similarly up to date. The FileZilla FTP server was the newest version. The SSH servers were OpenSSH 7.6p1, which is on the older side -- around two years old -- but there are no major CVE's for this server, so this is not an issue. The Microsoft Exchange server is version 15.1.1779.2, which is the most updated version for Windows Server 2016. The MediaWiki server was 1.32, which is a legacy version but was only released in January 2019. This is not a security risk.

The Apache server at 10.0.2.101 is still version 2.4.29, which is vulnerable to CVE-2019-0211. The Asterisk management system is version 5.0.1, which is out of date (it was using an outdated software naming convention). We do not know what version QueryTree is either because we do not have full access to the systems.

Every system that we can verify is strong except for the Apache server and the Asterisk management system. However, because we were unable to verify all software, we cannot give a rating of “Strong” and instead will give a rating of “Satisfactory”.

Related SIDs: SID-409

4.5 Management

Unsatisfactory

Organizations with effective management are resilient and have clear plans in place to address each component of the CIA triad: availability, integrity and confidentiality. They make good decisions that effectively mitigate risk. Management at these organizations can lead during times of stress and are able to recover from previous failures.

It was clear that DinoBank is severely lacking in many regards to the above criteria. We were unable to view how DinoBank responded when availability was an issue. When customer data was modified during the security assessment -- a clear violation of integrity – DinoBank was responded adequately. They worked with [redacted] to determine which customer was effected without violating their privacy and resolved the issue.

However, they have failed to respond to issues of confidentiality. During our previous assessment, there were many vulnerabilities discovered allowing unauthorized access to PII. A month has passed since then, and very few of these issues were resolved. There is evidence that DinoDank tried to resolve vulnerabilities with the PostgreSQL server and HTTP vs HTTPS protocols, but these efforts were not effective. Furthermore, DinoBank stores large amounts of PII in clear text, showing poor decision making in regards to confidentiality.

4.6 Audit Coverage

Strong

The Windows machines were analyzed for audit coverage because we had a high level of access to those systems. However, we were unable to assess the quality of logging on the Linux systems.

The Windows systems had effective, centralized logging that forwarded reports on the major threat surfaces. Splunk forwarders were installed on every machine and were used to forward logs to a single, centralized location. These included Powershell logs and Sysmon logs. Sysmon was installed using the SwiftOnSecurity configuration, a well known and effective configuration. Between the two of these, most exploitation pathways would make noise and generate alerts that could be viewed and anaylzed on Splunk.

The MediaWiki IIS server logged using default IIS logs. However, these logs were not forwarded to the Splunk management console.

The only exception was the FTP server, which did not have logging enabled. It appeared that Suricata, an IDS, was installed and intended to log network traffic, but it was not running.

Related SIDs: SID-402

5. Findings

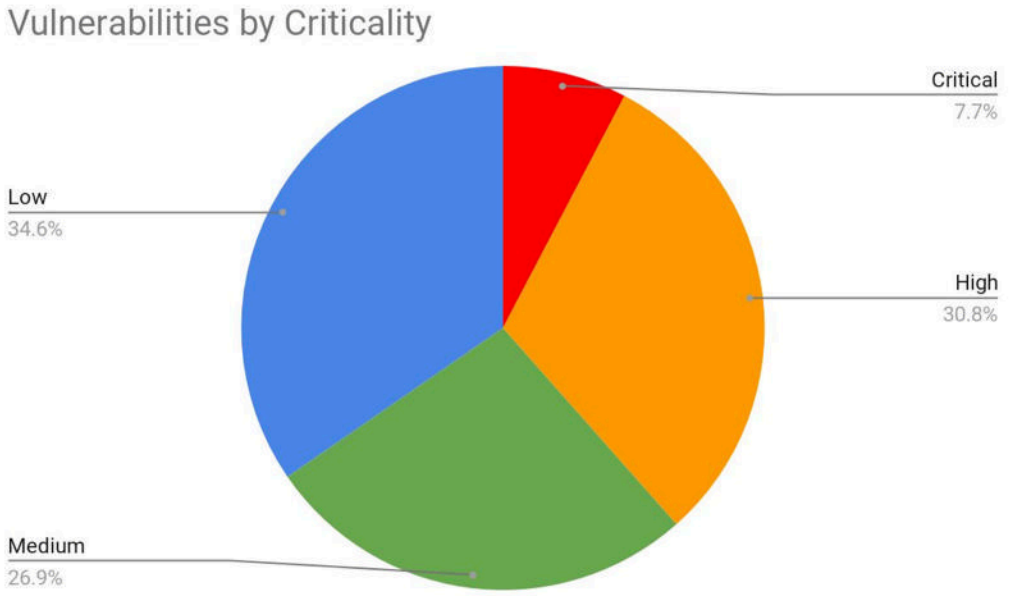


Figure 1: Vulnerabilities by Criticality

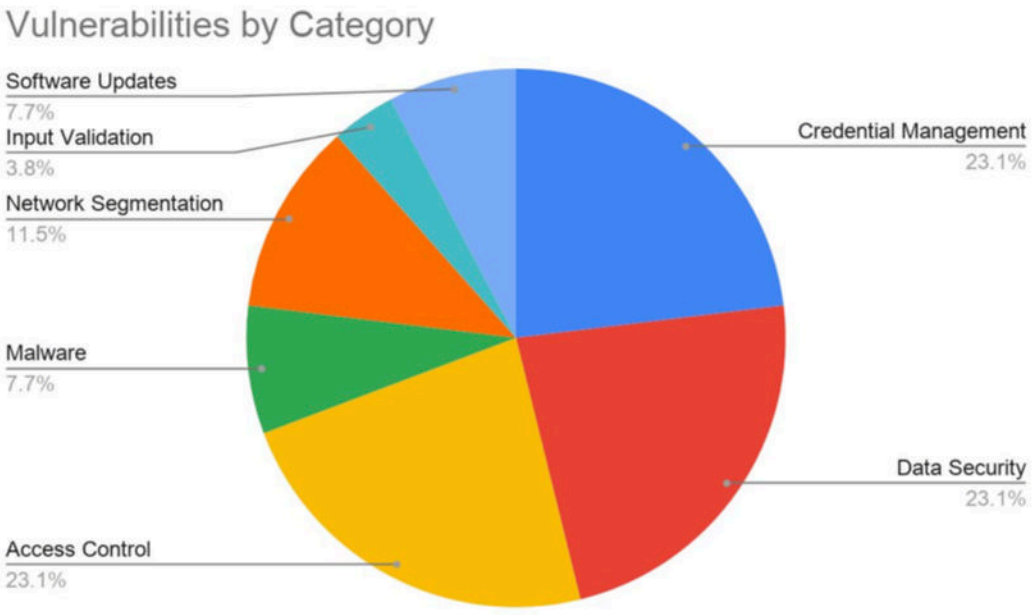


Figure 2: Vulnerabilities by Category

5.1 Critical Severity Findings

ID	Category	Title	Description
SID-101	Credential Management	PostgreSQL Metasploit Module	PostgreSQL access with default credentials
SID-102	Data Security	Unauthenticated money transfer	Attackers can transfer money from account to account without authentication.

5.1.1 Postgres Metasploit Module (SID-101)

Vulnerability ID: SID-101
Vulnerability Type: Data Security
Threat Level: Critical
CVE Number: CVE-2007-3280
MITRE ATT&CK Mitigation: M1030 Network Segmentation, M1036 Account Use Policies

Description:
The PostgreSQL database running on 10.0.2.100 allowed authentication with no username and password. This enabled us to use a metasploit module (CVE-2007-3280) to get code execution on the server as the Postgres user.

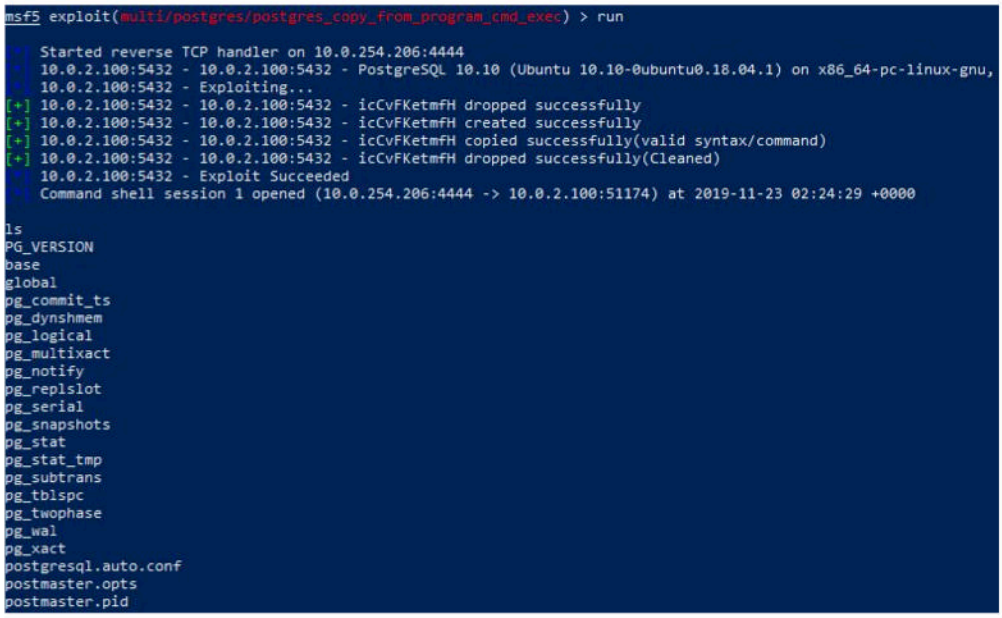


Figure 3

Steps to reproduce:

[redacted] used the Metasploit utility to exploit this vulnerability:

1. Use the 'auxiliary/admin/postgres/postgres_sql' module.
2. Set RHOST to 10.0.2.100, DATABASE to "", SQL to "SELECT datname FROM pg_database", and run the module. This will give you a list of valid database names, and you should see that one of the valid database names is 'indominusrex'
3. Use the 'exploit/multi/postgres/postgres_copy_from_program_cmd_exec' module.
4. Set RHOST to 10.0.2.100, DATABASE to `indominusrex`, and LHOST to the ip address of your own machine. Run the exploit, and you should have the shell.

Impact:

This level of access gave us unauthenticated access to the PostgreSQL database, which contained highly valuable information. It included sensitive financial information for all customers, as well as social security numbers and PIN numbers. In addition, the employee database contained emails, usernames, and passwords, which may be used in other locations across the network to login to other machines and services. All of these information are stored in plain text, which gives attackers direct access to the sensitive information.

Figure 4

Recommendation:
Change the credentials of the postgres database to be a non-default credential. Sensitive information such as PII should be hashed instead of stored in plain text.

5.1.2 Unauthenticated Money Transfer (SID-102)

Vulnerability ID: SID-102
Vulnerability Type: Data Security
Threat Level: Critical
MITRE ATT&CK Mitigation: M1054 Software Configuration

Description:
Attackers could browse to https://my.dinobank.us/transfer.php to transfer money from one account to another without authentication. Attackers only need to know their own account number to transfer money to their account, as they could choose to not specify a sender. The application will pick a designated account as the sender, which will allow the attacker to gain arbitrary money at the expense of the designated account. Finally, if the attacker puts their own account number on both fields (or leaves both fields blank), then the attacker could gain arbitrary amounts of money from DinoBank (no money is subtracted from any other customer).

Steps to reproduce:

1. Use a directory enumeration tool to discover that /transfer.php is a valid page in the web application
2. Visit the page, specify the receiver's account number and the amount of money to transfer (or only specify the amount of money to transfer), and submit the form.

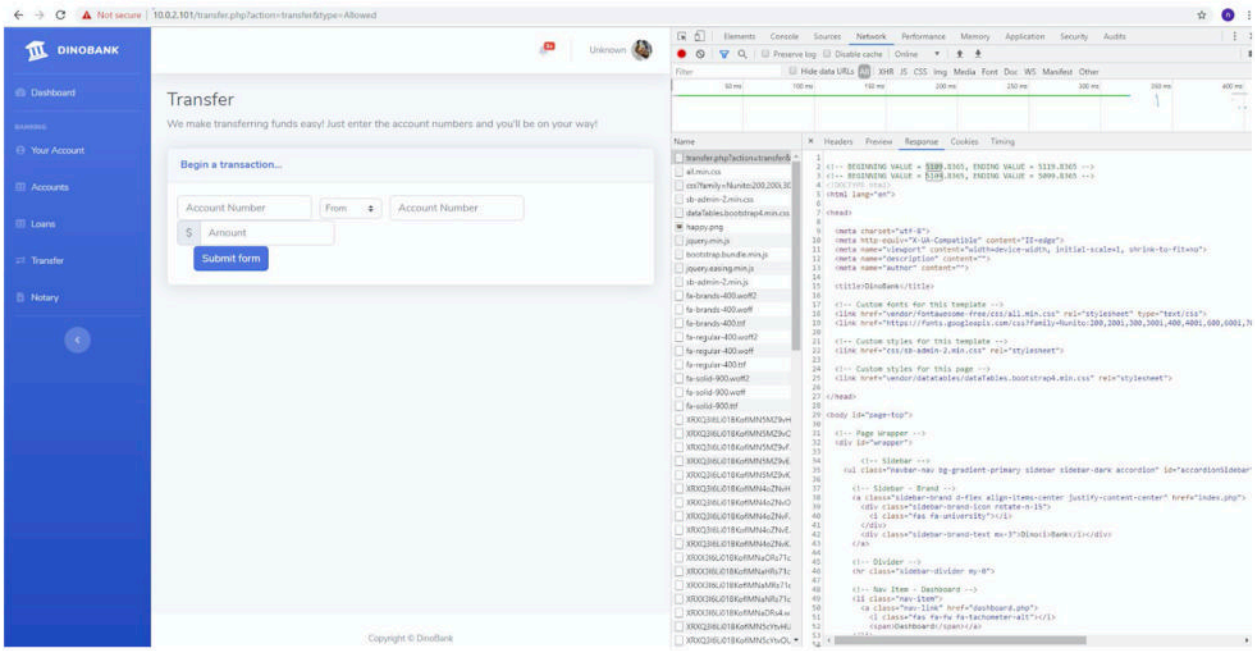


Figure 5

Impact:

This vulnerability could compromise the integrity of the bank balances of customers, allowing them to take money from other customers and from Dinobank itself.

Recommendation:

The /transfer.php page should be configured to require a valid PHPSESSID cookie before redirecting users to the money transfer page, otherwise redirect users to the login page. Additionally, ensure that the sender's account number is not the same as the receiver's account number in the form.

5.2 High Severity Findings

ID	Category	Title	Description
SID-201	Credential Management	Credentials on Internal Wiki	Cleartext credentials on internal wiki
SID-202	Credential Management	Password Reuse	Passwords being reused across the network
SID-203	Credential Management	Weak Passwords	Insufficiently secure passwords were in use
SID-204	Access Control	FileZilla Anonymous FTP	Unauthenticated users had read access to root of FTP server
SID-205	Credential Management	Configuration Scripts	Leftover provisioning scripts included plaintext passwords
SID-206	Data Security	Passwords Stored in Plaintext	Files containing privileged passwords were stored on desktops
SID-207	Data Security	Data in SYSVOL Share	Significant amount of company files openly shared with Everyone
SID-208	Network Segmentation	Unnecessarily Exposed Services	A number of services were left exposed

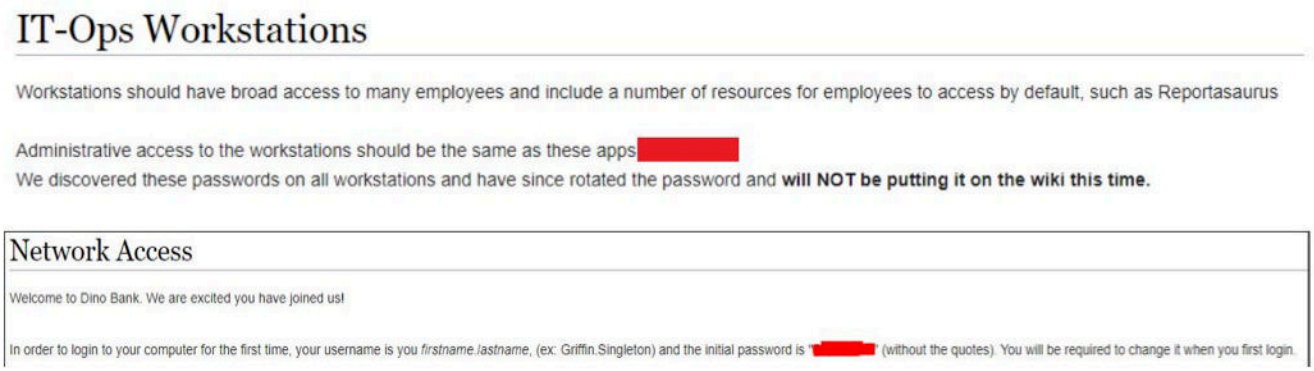
5.2.1 Credentials on Internal Wiki (SID-201)

Vulnerability ID: SID-201
Vulnerability Type: Credential Management
Threat Level: High
MITRE ATT&CK Technique: T1087 - Account Discover
MITRE ATT&CK Mitigation: M1035 - Limit Access to Resource Over Network

Description:

On the internal wiki hosted at 10.0.1.31, there was a webpage called “IT-Ops Workstations” which discussed how to log in as Administrator on workstations.

Furthermore, a page called "DinosRUs!" contains the default password for all users. It can be viewed with no account or credentials of any kind necessary. It mentioned that the old password "Password1" was rotated and that the new password would not be placed on the wiki.



IT-Ops Workstations

Workstations should have broad access to many employees and include a number of resources for employees to access by default, such as Reportasaurus

Administrative access to the workstations should be the same as these apps[redacted]
We discovered these passwords on all workstations and have since rotated the password and **will NOT be putting it on the wiki this time.**

Network Access

Welcome to Dino Bank. We are excited you have joined us!

In order to login to your computer for the first time, your username is you *firstname.lastname*, (ex: Griffin.Singleton) and the initial password is "[redacted]" (without the quotes). You will be required to change it when you first login.

Figure 6

Steps to reproduce:
Go to "http://10.0.1.31/index.php?title=IT-Ops_Workstations" to see the admin password message and "http://10.0.1.31/index.php?title=Network_Access to see the default password.

Impact:
This wiki page is exposed to the entire world and requires no credentials to view. Thus, any clever attacker can get these credentials and use them to access workstations across the DinoBank network. These workstations are connected to the DinoBank Windows domain, so they can be used to pivot and spread through the entire network. Thus, this vulnerability can, with moderate effort, be used to gain domain-wide Windows access and exfiltrate lots of company information. Some simple guessing led us to find the new administrator password, which allowed us to log in as Administrator on all branch workstations.

Recommendation:
The wiki server at 10.0.1.31 should be hidden behind a firewall. Wiki pages should only be viewable when logged in with valid AD credentials. The workstation Administrator accounts should be changed such that they are not all the same password (see

SID-202), and to use stronger passwords that are at least 12 characters in length consisting of multiple words, special characters, and numbers (see SID-203).

5.2.2 Password Reuse (SID-202)

Vulnerability ID: SID-202
Vulnerability Type: Credential Management
Threat Level: High
MITRE ATT&CK Mitigation: M1027 Password Policies

Description:
Multiple passwords throughout DinoBank were reused throughout the network. We noticed the following passwords being reused:

- Local Administrator Passwords on Branch Workstations and Tellers
- Default Passwords for Domain Users

Impact:
By reusing passwords, this has the effect that if one password or system is compromised, the attacker can use the acquired passwords to spread throughout the network and gain further access. For new employees and/or accounts, using the same new password or even procedurally generated passwords may allow attackers to be able to compromise new accounts.

Recommendation:
We recommend all accounts have their passwords changed to be unique. We also recommend randomly generating default passwords for new employees. One software that could be used to implement this is Microsoft's Local Administrator Password Solution (LAPS) which we highly recommend deploying.

5.2.3 Weak Passwords (SID-203)

Vulnerability ID: SID-203
Vulnerability Type: Credential Management
Threat Level: High

MITRE ATT&CK Mitigation: M1027 Password Policies

Description:
Many systems and services across the network were utilizing incredibly insecure passwords. Beyond that, all passwords recovered not contained within the PostgreSQL database consisted of words relating to dinosaurs or banking, with occasional letters swapped out for numbers or symbols.

Impact:
By guessing passwords, we were able to gain local administrator access to all workstations on the branch networks (ten computers).

Recommendation:
We recommend ensuring all passwords either are at least 12 random characters or utilize passphrases of at least 20 characters. Furthermore, 2FA should be required for all accounts through a service such as RSA SecurID.

5.2.4 FileZilla Anonymous FTP (SID-204)

Vulnerability ID: SID-204
Vulnerability Type: Access Control
Threat Level: High
MITRE ATT&CK Mitigation: M1035 Limit Access to Resource Over Network

Description:
The FileZilla FTP server on 10.0.1.12, which is accessible from outside the DinoBank subnets, allowed for anonymous login, and allowed users to access the entire C: drive. Normally, these vulnerabilities would be listed at 'Medium' criticality, but combined they give unauthenticated read access to the entire C: drive of the server and are thus a 'High' vulnerability.

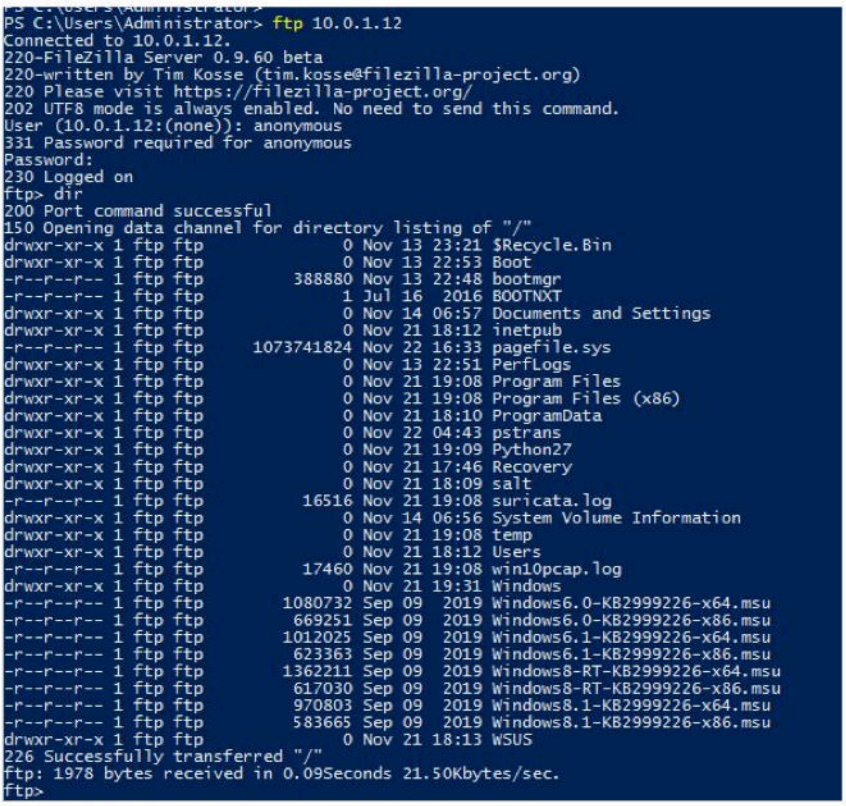


Figure 7

Steps to reproduce:

1. Connect over FTP to 10.0.1.12
2. When prompted for the ‘username’, say ‘anonymous’.
3. When prompted for the ‘password’, say anything.
4. Once you have connected, type ‘dir’ to see the entire C: drive.

Impact:

The vulnerabilities give unauthorized read access to the entire C: drive. This allowed us to read Powershell scripts in the C:\pstrans directory which were used to configure the machine. The contained Domain Administrator credentials, which eventually allowed us to spread through the entire Active Directory.

Additionally, it allowed us to see settings for FileZilla, Splunk, WSUS, and any user files. These could be used to allow malicious adversaries to learn about monitoring technologies in the DinoBank network and then move quietly through the network.

Recommendation:
The FTP server should not be visible from outside the DinoBank subnets. Local firewall rules can be used on both the FTP and Administrator-FTP ports to limit this network access. Anonymous FTP should then be disabled in the FileZilla Administrator console. Finally, the FTP server should be configured to only give access to a single directory with no critical files. It appears that there is a C:\temp directory intended for this purpose, but the server configuration is not correct.

5.2.5 Passwords in PowerShell Script Block Logging (SID-205)

Vulnerability ID: SID-205
Vulnerability Type: Credential Management
Threat Level: High
MITRE ATT&CK Mitigation: M1043 Credential Access Protection

Description:
In the pstrans directory of numerous Windows workstations, powershell logs containing domain administrator credentials were accessible. This includes 10.0.1.12 and all of the workstations that we accessed. The Powershell commands contained Base64 encoded commands, and some of these commands contained credentials for Domain accounts. On 10.0.1.12, one of these scripts contained credentials for Domain Administrator accounts. The decoded portion of the script is shown below.

```
if (Test-Path variable:global:ProgressPreference){$ProgressPreference="SilentlyContinue"}
$f = $s.GetFolder("\")
$f.RegisterTaskDefinition($name, $t, 6, "Administrator", '[redacted]', 1, $null) | Out-Null
$t = $f.GetTask("\$name")
$t.Run($null) | Out-Null
```

Figure 8

Impact:
This allowed for escalation from local user to Domain Administrator. This simple escalation of privileges makes it trivial to spread through the entire Active Directory network.

Recommendation:
These logs should be deleted across all Windows machines. Furthermore, no files, logs, or scripts should contain any plaintext passwords.

5.2.6 Passwords Stored in Plaintext File (SID-206)

Vulnerability ID: SID-206
Vulnerability Type: Password Policies
Threat Level: High
MITRE ATT&CK Mitigation: M1027 - Password Policies

Description:

Numerous systems contained plaintext passwords in text documents on the Administrator account's desktop. This provided access to Kracken, Reportasaurus, Wiki, Core/Bankweb, Coin Heads, Coin Tails, the Ethereum Croissant Coin contract, and QueryTree.

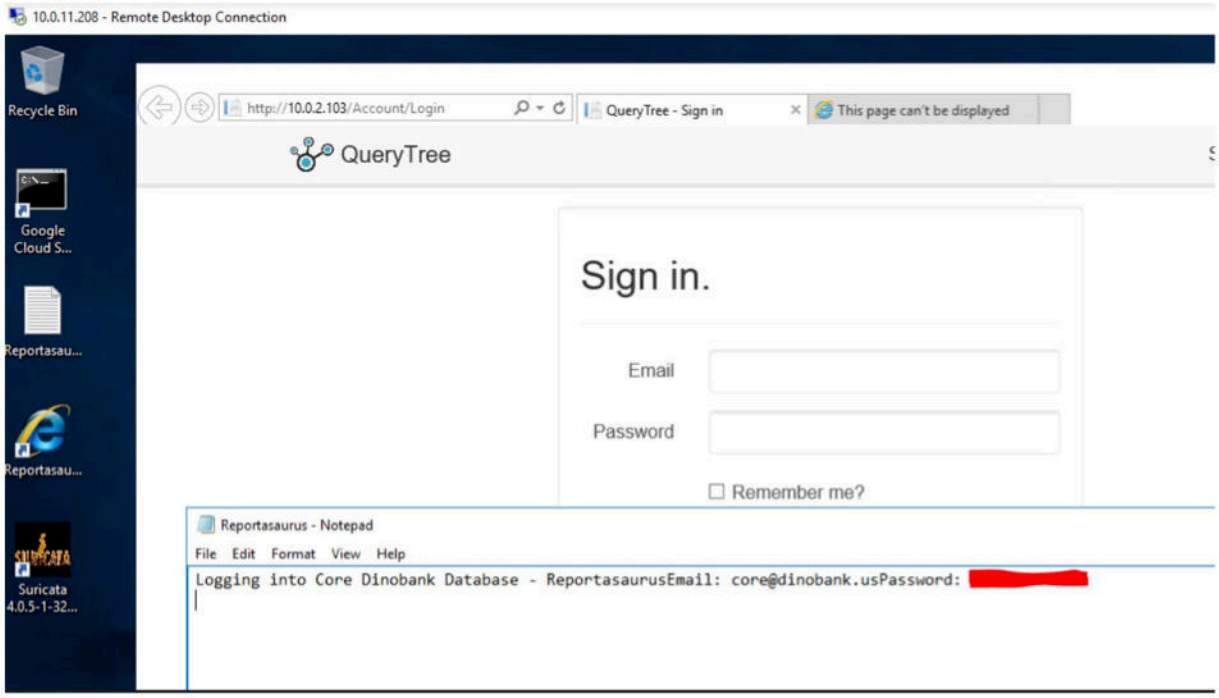


Figure 9

Figure 10

Impact:
By storing these credentials in plaintext files, attackers that gain access to one system could use these passwords to gain a high level of access across the entire network.

Recommendation:
Do not store any passwords in plaintext files on any system.

5.2.7 Data in SYSVOL Share (SID-206)

Vulnerability ID: SID-206 Vulnerability Type: Data Security Threat Level: High MITRE ATT&CK Mitigation: M1035 Limit Access to Resource Over Network

Description:

On the CORP Domain Controller, there was an open share in SYSVOL/departments/ that contained shared files among each department in the company. These shared contained documents named such as "PayStatement.pdf," "UserList.xlsx," and "workstations.csv." Many of these documents seemed to contain unrelated wiki-style information, but they hinted that they contain company intellectual property or otherwise sensitive information.

Impact:
Although it is common practice in corporate environments to store files on folder shares, all users had access to every department's files. This lack of access control on a per user basis meant that, for example, an employee working in accounting could access IT diagrams or HR information, despite not having a business case to do so.

Recommendation:
We recommend utilizing Active Directory Security Groups to organize users by department. Once everyone is correctly categorized into these groups, you can limit each user's access to only their department's shared files. We also recommend performing an audit to remove unneeded files in this share as it represents a centralized point of interest for attackers to search for sensitive company information.

5.2.8 Unnecessarily Exposed Services (SID-207)

Vulnerability ID: SID-207 Vulnerability Type: Network Segmentation Threat Level: High MITRE ATT&CK Mitigation: M1035 Limit Access to Resource Over Network, M1030 Network Segmentation

Description:
A significant majority of all services throughout the network had no real need to be exposed to the open internet, but there was no firewall in place to regulate this. A list of systems that should have been allowed internal network communication but no external communication is below.

- All SMB and RPC across the Windows network. RDP may be allowed, but this should be through a VPN rather than directly.
- All domain controller services such as LDAP and Kerberos.

- SSH may be allowed, but this should be through a VPN rather than directly.
- PostgreSQL core database and API
- Exchange admin portal may be allowed, but this should be through a VPN rather than directly
- FTP
- All services related to the internal wiki

Impact:
Nearly all access we were able to obtain was through exposed services that did not need to be accessible on the open internet. With properly whitelisted ports, we would have been able to gain only little if any access to the DinoBank system. This is one of the single most impactful changes DinoBank can make to their system.

Recommendation:
We recommend investing in a network firewall for DinoBank's network and whitelisting only ports that need remote access such as customer-facing services. In addition to a network firewall, all systems should use a host-based firewall (iptables on linux or netsh on Windows) to ensure proper network segmentation. Furthermore, DinoBank should configure a VPN to facilitate access to systems and services that employees may need to access remotely but should not be accessed by customers.

5.3 Medium Severity Findings

ID	Category	Title	Description
SID-301	Access Control	Media Wiki Edit Permissions	Users with no accounts can edit the internal wiki page
SID-302	Access Control	Anonymous SAM enumeration	Branch Domain controllers allowed anonymous user enumeration.
SID-303	Malware	Malware Discovered	Malware was discovered on DinoBank systems
SID-304	Credential Management	Unenforced and Weak Password Policies	Password policies were weak and not enforced
SID-305	Network Segmentation	Local Firewalls Disabled	Windows machines had their local Windows Firewall disabled
SID-306	Malware	Malicious Scheduled Tasks	Multiple scheduled tasks appeared to misconfigure Windows
SID-307	Input Validation	XSS On Trading Website	Cryptocurrency Trading Website has XSS in posts

5.3.1 MediaWiki Edit Permissions (SID-301)

Vulnerability ID: SID-301
Vulnerability Type: Access Control
Threat Level: Medium
MITRE ATT&CK Mitigation: M1022 Restrict File and Directory Permissions

Description:
The internal Media Wiki at 10.0.1.31 is editable even when not logged in to an account. Because this wiki is exposed to the entire Internet, this means anyone in the world could edit the wiki for whatever purpose.

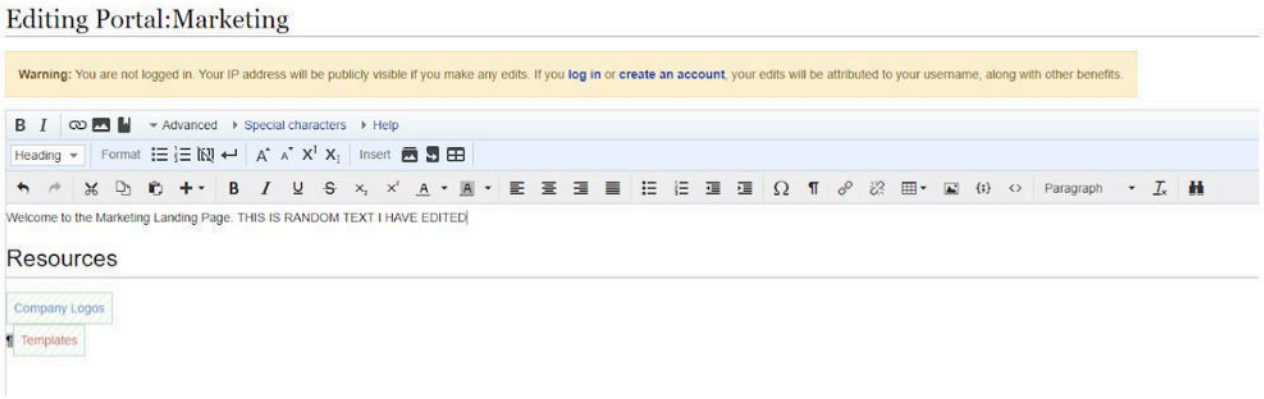


Figure 11

Steps to reproduce:

Go to any wiki page at 10.0.1.31. In the menu at the top-right of the page, there is a button that says 'Edit'. Once clicked, it will redirect the user to the page to edit the wiki.

Impact:

This could allow attackers to change hyperlinks on Wiki pages to go to their malicious websites. Because the link would come from a company website, it may be perceived as being official. If the link goes to a malicious phishing page, employees or customers might be likely to ignore any suspicious elements, thus resulting in a highly effective phishing campaign that steals lots of PII.

Recommendation:

The wiki server should be hidden behind a firewall so it is not accessible from outside the company subnets. Permissions should be configured such that only authenticated users can view or edit wiki pages, and the authentication should require valid AD accounts.

5.3.2 Anonymous SID/Name Translation (SID-302)

Vulnerability ID: SID-302
Vulnerability Type: Access Control
Threat Level: Medium
MITRE ATT&CK Technique: T1087 Account Discovery
MITRE ATT&CK Mitigation: M1015 Active Directory Configuration

Description:
The branch domain controllers allowed anonymous SID/Name Translation over SMB. This allowed [redacted] to run a scanner against the domain controllers to gain a complete list of all users and service accounts on the network using the network's SID and the default RID range of users.

Steps to reproduce:
In metasploit, `use auxiliary/scanner/smb/smb_lookupsid` then `set RHOSTS` to 10.0.10.100, 10.0.11.100, or 10.0.12.100`. `run` the module, and it will output a list of users.

Impact:
With a list of valid users, an attacker could then begin a password spraying campaign. Given the relative weakness of a large portion of passwords, this would likely allow a malicious attacker to break passwords within a few days.

Recommendation:
Change the group policy "Network access: Allow anonymous SID/Name translation" under "Computer Configuration\Windows Settings\Security Settings\Local Policies\ Security Options" to disabled.

5.3.3 Malware Discovered (SID-303)

Vulnerability ID: SID-303 Vulnerability Type: Malware Threat Level: Medium MITRE ATT&CK Technique: T1060 Registry Run Keys / Startup Folder MITRE ATT&CK Mitigation: M1049 Antivirus/Antimalware

Description:
During the engagement, [redacted] identified what appears to be a piece of malware called "miner.exe". This executable was set to run on startup through a registry run key on every non-domain controller system. The exact source of this executable is not entirely clear as it was installed before logging was present on the systems. The owner of the executable appeared to be the local administrator, and the creation times for the executables across the network were close together. This indicates that it was

likely installed as part of the provisioning of the network and therefore attributable to one or more employees.

This executable was uploaded to VirusTotal, where it had no detections. As part of Hybrid Analysis's service, it was run through an AI-based detection and determined to be malware. This executable was written in C#, so a more thorough analysis should be able to extract the source and determine the exact purpose, though a cursory look indicates that it is likely a cryptocurrency miner. Analyzing the source further would likely be able to attribute the owner given that it must have the owner's cryptocurrency wallet address.

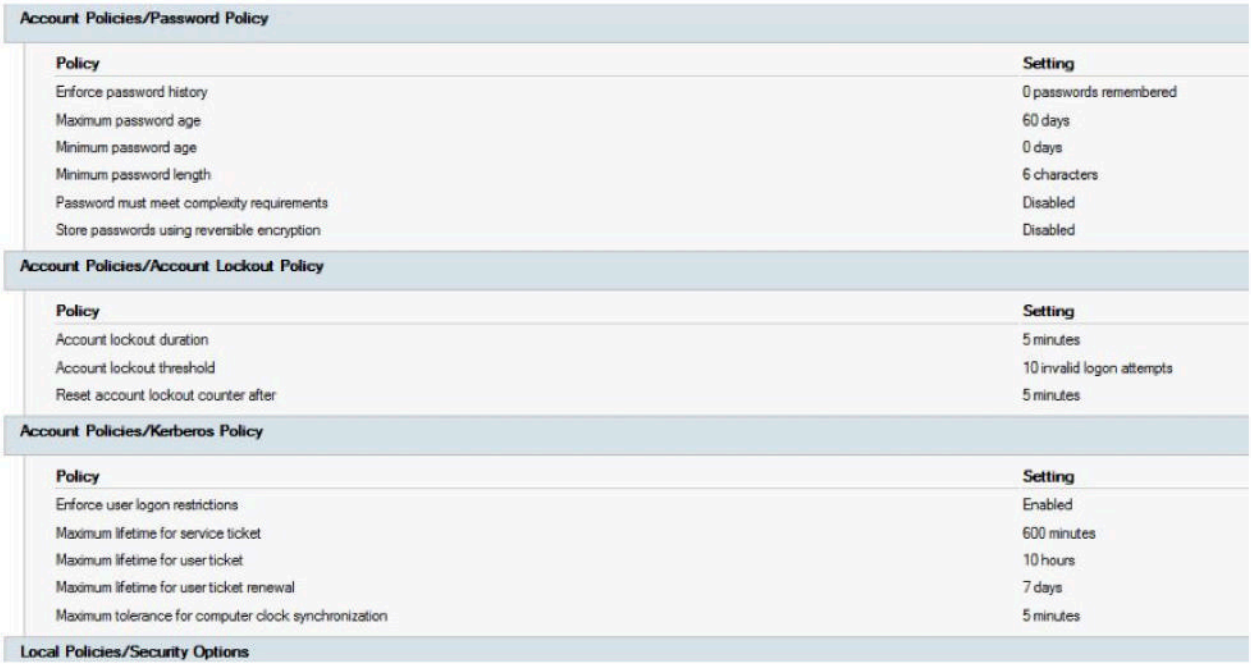
Impact:
During the engagement, there were several complaints regarding slow services on services [redacted] was not testing at the time. Though it is difficult to confirm, this executable may have been part of the cause. Furthermore, the possibility that there is an insider threat could have far-reaching consequences that could ultimately end up affecting service uptime, data integrity, and more.

Recommendation:
Reverse engineer the binary to attribute it to its creator and installer, remove the autorun, and then delete the executable and its run key.

5.3.4 Weak and Unenforced Password Policies (SID-304)

Vulnerability ID: SID-304 Vulnerability Type: Credential Management Threat Level: Medium MITRE ATT&CK Technique: T1110 Brute Force MITRE ATT&CK Mitigation: M1027 Password Policies

Description:
Upon reviewing the password policies, we determined that they were wholly insufficient for a production business environment. Not only were password requirements lacking; they were not even enforced. The password age set was reasonable, but having zero passwords remembered completely nullifies any security benefit from a maximum password age.



Account Policies/Password Policy

Policy	Setting
Enforce password history	0 passwords remembered
Maximum password age	60 days
Minimum password age	0 days
Minimum password length	6 characters
Password must meet complexity requirements	Disabled
Store passwords using reversible encryption	Disabled

Account Policies/Account Lockout Policy

Policy	Setting
Account lockout duration	5 minutes
Account lockout threshold	10 invalid logon attempts
Reset account lockout counter after	5 minutes

Account Policies/Kerberos Policy

Policy	Setting
Enforce user logon restrictions	Enabled
Maximum lifetime for service ticket	600 minutes
Maximum lifetime for user ticket	10 hours
Maximum lifetime for user ticket renewal	7 days
Maximum tolerance for computer clock synchronization	5 minutes

Local Policies/Security Options

Figure 12

Impact:

With the current rules in place, employees could have horribly insecure passwords. Furthermore, with 10 invalid login attempts being allowed every five minutes, an attacker could spray thousands of passwords per day. Given the weak requirements, this makes credential spraying a feasible method of access for an attacker.

Recommendation:

First and foremost, while the current maximum password age is reasonable, a thirty day maximum age would be an improvement. Modern standards state that at least five previous passwords should stored. Passwords should be at least twelve characters long, and passwords should be required to meet complexity requirements. The lockout duration and reset counter should both be increased to thirty minutes, and the number of invalid logins should be decreased to three.

5.3.5 Firewall Disabled (SID-305)

Vulnerability ID: SID-305
Vulnerability Type: Network Segmentation
Threat Level: Medium
MITRE ATT&CK Mitigation: M1037 Filter Network Traffic

Description:
During our engagement, we obtained access to a number of Windows workstations and servers that had their Windows Firewall turned off. Although this is somewhat common, host based firewalls allow administrators to properly lock down host to host communication as part of a good defense in depth strategy.

Impact:
These machines do not have any network level filtering at the host level which means they allow traffic from any port to any port. Additionally, it also means these machines have no restrictions on what they can communicate to. For example, many servers do not require access to the internet for them to function, and this access could be turned off via firewall rules.

Recommendation:
At a minimum, the Windows Firewall should be enabled on all computers. It can be enabled and pushed out to all DinoBank systems through group policy. Default Windows Firewall rules will be a good start, although we recommend staged rollout to ensure that this does not affect system connectivity. Additionally, we recommend beginning to tailor specific firewall configurations to specific servers to further limit and control traffic when your network segmentation policies is more mature.

5.3.6 Malicious Scheduled Tasks (SID-306)

Vulnerability ID: SID-306 Vulnerability Type: Malware Threat Level: Medium MITRE ATT&CK Technique: T1053 Scheduled Task

Description:
During our engagement, we identified a number of scheduled tasks on Windows systems that appeared to be deliberately applying insecure policies, from disabling the firewall to weakening encryption to disabling antivirus.

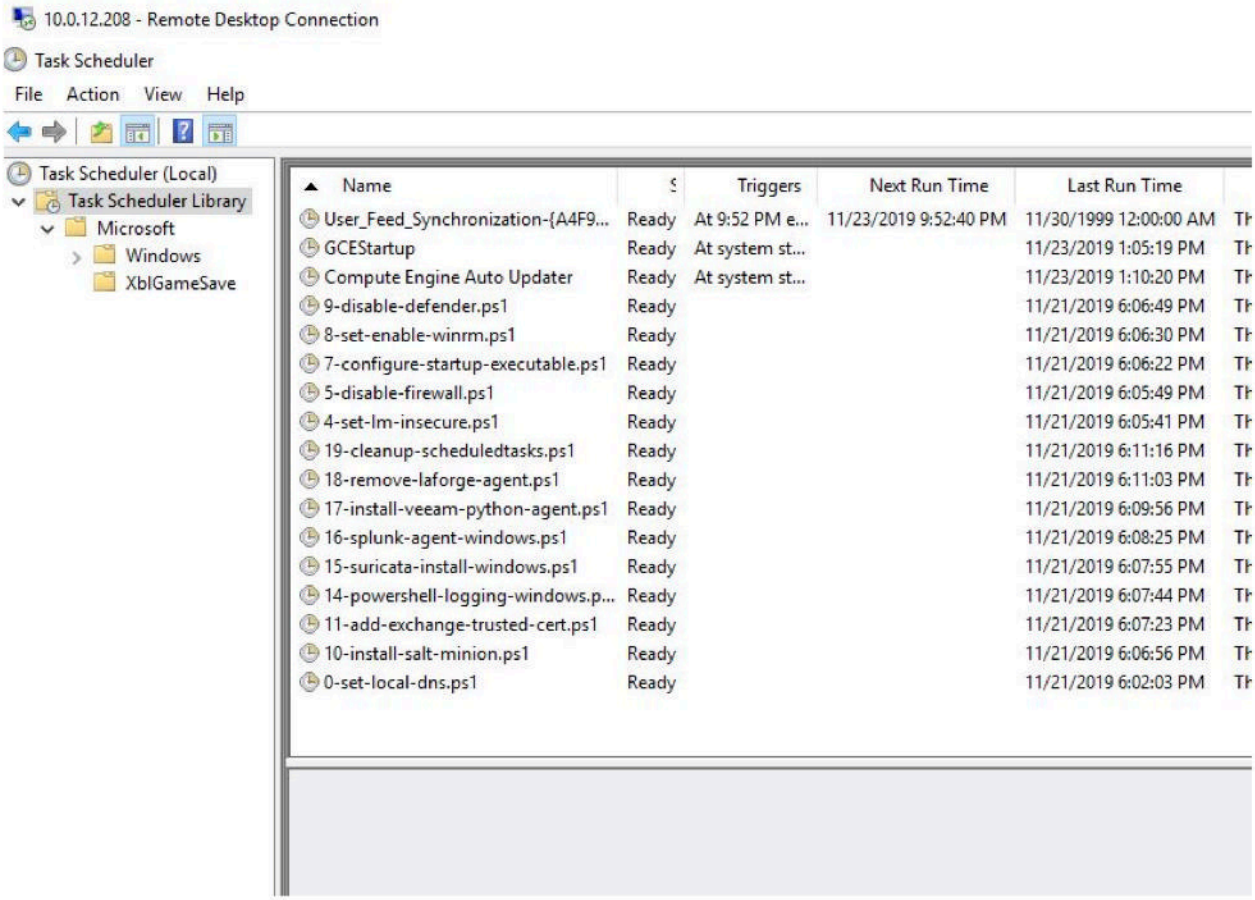


Figure 13

Impact:

These actions drastically reduce the overall security of the network and make it significantly easier for malicious actors to gain credentials and access to systems. Furthermore, such tasks are indicative of more malware present on the system or an insider threat, both of which are extremely serious issues.

Recommendation:

Perform an immediate log assessment to determine the source of these malicious scheduled tasks. It is possible that the file system link for these files was removed but the memory still exists, in which case recovering these malicious powershell scripts is possible. Finally, remove these scheduled tasks across the network and repair the settings they changed.

5.3.7 XSS on Trading Site (SID-307)

Vulnerability ID: SID-307 Vulnerability Type: Input Sanitization Threat Level: Medium MITRE ATT&CK Technique: T1064 Scripting

Description:

Trade advertisements found on the cryptocurrency trading website at 10.0.1.250 are vulnerable to an XSS attack by embedding script tags in the terms of the trade.

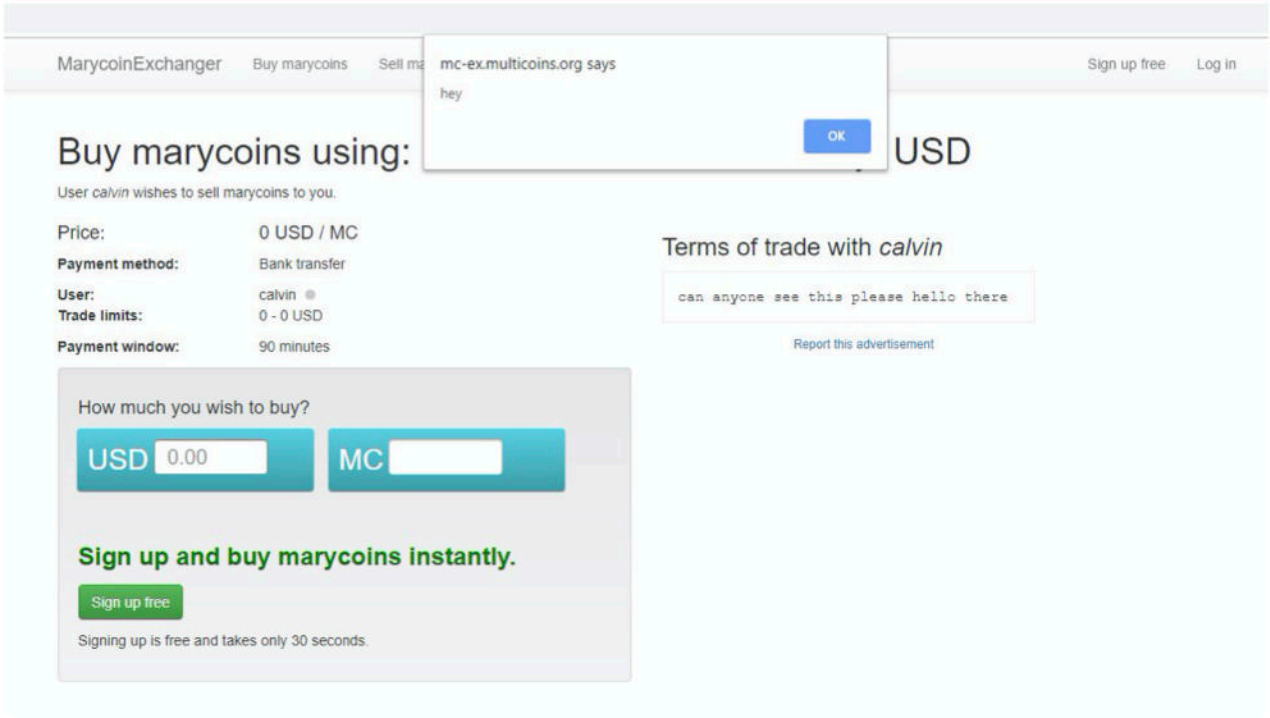


Figure 14

Steps to reproduce:

When creating a trade, in the “Terms of trade” field, input:

```
<script>alert(“My XSS attack”);</script>
```

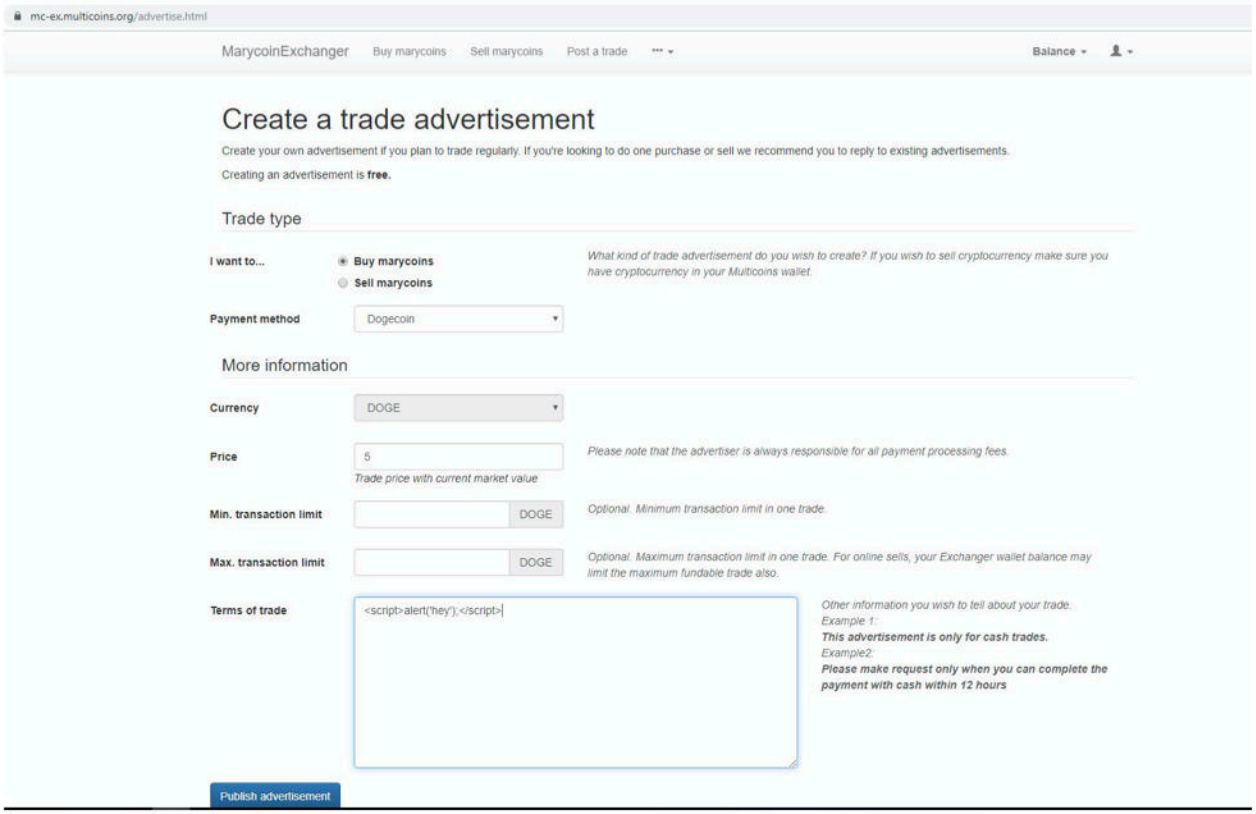


Figure 15

When the details of the trade are viewed, there should be a pop-up that appears with the text “My XSS attack”.

Impact:

Since the web application uses cookies to keep track of logins, attackers can use the XSS to steak cookies and log in as the victim. They could then access their account on the trading website, and transfer cryptocurrencies from the victim’s account to their own. While this is not an official DinoBank website, it is hosted on DinoBank servers and this issue would reduce customer trust in the bank.

Recommendation:

The data should be sanitized to prevent JavaScript execution.

5.4 Low Severity Findings

ID	Category	Title	Description
SID-401	Access Control	MediaWiki Account Creation	Any person can create account for company wiki
SID-402	Data Security	No logging on Filezilla FTP	Lack of logging for FTP service
SID-403	Access Control	FileZilla Bounce Attack	Allows unauthenticated users to perform port scans on internal network
SID-404	Data Security	Improper TLS implementation	Web Servers were not properly configured for HTTPS
SID-405	Network Segmentation	IVR Management Server Exposure	Management Server API was directly accessible
SID-406	Access Control	Anonymous share enumeration	Anonymous users could enumerate SMB shares
SID-407	Data Security	Ether web API source code disclosure	Users could view the source code of the ether API.
SID-408	Software Updates	Usages of SMBv1	All Windows systems were using SMBv1
SID-409	Software Updates	Outdated Apache Server Version	Network scans revealed a web server running an outdated version of Apache.

5.4.1 Media Wiki Arbitrary Account Creation (SID-401)

Vulnerability ID: SID-401 Vulnerability Type: Access Control

Threat Level: Low
MITRE ATT&CK Mitigation: M1036 Account Use Policies

Description:
The Media Wiki server at 10.0.1.31 allows accounts to be created with any username and password. This means that one does not need to be a DinoBank employee with legitimate DinoBank credentials to create an account and use the Media Wiki server.

Impact:
Even if the wiki is patched so actors who are not logged in are no longer able to view or edit pages, it would be a minimal barrier. They could still easily create an account despite knowing nothing about DinoBank and then proceed to learn about DinoBank infrastructure or edit wiki pages. They could then use the wiki as described in "SID-301: Media Wiki Edit Permissions" to execute highly effective phishing schemes.

Recommendation:
Authentication on the Media Wiki server should be linked to the Windows AD authentication such that having a Windows account in the DinoBank domain allows one to log in to the wiki, and is the only way to log in to the wiki.

5.4.2 No logging on Filezilla FTP (SID-402)

Vulnerability ID: SID-402
Vulnerability Type: Data Security
Threat Level: Low
MITRE ATT&CK Mitigation: M1047 - Audit

Description:
The FileZilla FTP server hosted at 10.0.1.12 did not log any information.

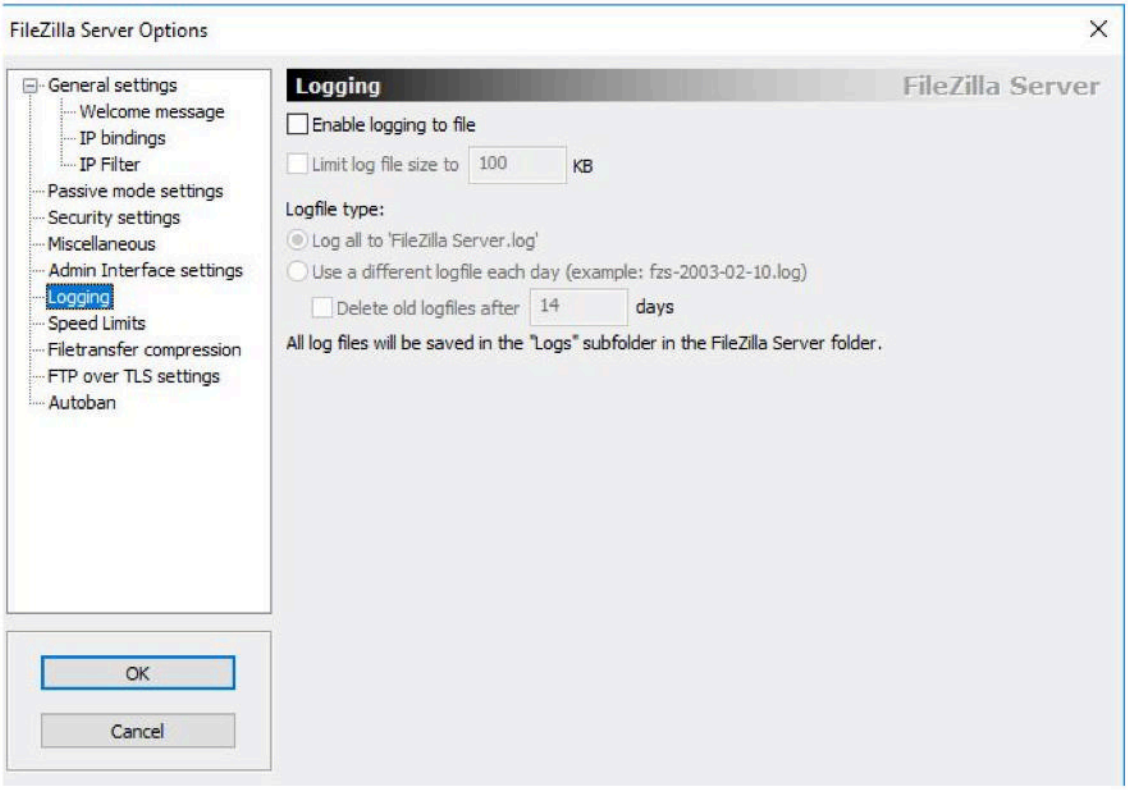


Figure 16

Impact:

If malicious actors were attempting to brute force FTP credentials or were successful and able to access the server, there would be no way for DinoBank administrators to know. This could potentially allow adversaries to gain access to the network unnoticed.

Recommendation:

In the FileZilla Server Options menu shown above, go to 'Logging' and check 'Enable logging to file'. These logs should then be backed by Splunk so they can be accessed from the Splunk management console and analyzed with Splunk searching tools.

5.4.3 FileZilla Bounce Attack (SID-403)

Vulnerability ID: SID-403
Vulnerability Type: Access Control
Threat Level: Low
MITRE ATT&CK Mitigation: M1051 Update Software

Description:

The FileZilla FTP was vulnerable to bounce attacks, which could allow unauthenticated users to do port scans on internal hosts.

```
/envs/nationals-cptc [redacted]/kali01 @~ # nmap -sC -sV 10.0.1.12 -p 21
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-23 14:55 UTC
Nmap scan report for nationals-t4-corp-corp-wsus-01.c.infra-test-environment.internal (10.0.1.12)
Host is up (0.0014s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     FileZilla ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x 1 ftp ftp              0 Nov 13 23:21 $Recycle.Bin
| drwxr-xr-x 1 ftp ftp              0 Nov 13 22:53 Boot
| -r--r--r-- 1 ftp ftp         388880 Nov 13 22:48 bootmgr
| -r--r--r-- 1 ftp ftp              1 Jul 16  2016 BOOTNXT
| drwxr-xr-x 1 ftp ftp              0 Nov 14 06:57 Documents and Settings
| drwxr-xr-x 1 ftp ftp              0 Nov 21 18:12 inetpub
| -r--r--r-- 1 ftp ftp     1073741824 Nov 23 13:05 pagefile.sys
| drwxr-xr-x 1 ftp ftp              0 Nov 13 22:51 PerfLogs
| drwxr-xr-x 1 ftp ftp              0 Nov 21 19:08 Program Files
| drwxr-xr-x 1 ftp ftp              0 Nov 21 19:08 Program Files (x86)
| drwxr-xr-x 1 ftp ftp              0 Nov 21 18:10 ProgramData
| drwxr-xr-x 1 ftp ftp              0 Nov 23 13:05 pstrans
| drwxr-xr-x 1 ftp ftp              0 Nov 21 19:09 Python27
| drwxr-xr-x 1 ftp ftp              0 Nov 21 17:46 Recovery
| drwxr-xr-x 1 ftp ftp              0 Nov 21 18:09 salt
| -r--r--r-- 1 ftp ftp          16516 Nov 21 19:08 suricata.log
| drwxr-xr-x 1 ftp ftp              0 Nov 14 06:56 System Volume Information
| drwxr-xr-x 1 ftp ftp              0 Nov 21 19:08 temp
| drwxr-xr-x 1 ftp ftp              0 Nov 21 18:12 Users
| -r--r--r-- 1 ftp ftp          17460 Nov 21 19:08 win10pcap.log
|_Only 20 shown. Use --script-args ftp-anon.maxlist=-1 to see all.
|_ftp-bounce: bounce working!
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.28 seconds
```

Figure 17

Steps to reproduce:

On a machine with nmap, execute the following script:

`nmap -sC -sV <IP> -p 21`

The bottom of the output, before the 'Service Info', will state if the server is vulnerable to ftp-bounce attacks.

Impact:

Since this FTP server had anonymous read access to C:\, scans on the internal service were mostly unnecessary. However, if there was less access to 10.0.1.12 the FTP bounce attack could allow attackers to send data to internal services.

Recommendation:

DinoBank should either disable anonymous access to FileZilla, or hide it behind a firewall so external attackers can't access it.

5.4.4 Improper TLS Implementation on Web Servers (SID-404)

Vulnerability ID: SID-404
Vulnerability Type: Data Security
Threat Level: Low
MITRE ATT&CK Mitigation: M1020 SSL/TLS Inspection

Description:

Several web servers lacked proper TLS implementation. This prevents information transmitted between the client and server from being encrypted. This applied to several key DinoBank websites, including the customer banking portal located at 10.0.2.101 and the QueryTree application located at 10.0.2.103.

Steps to reproduce:

For the customer banking portal, simply open a browser, type 10.0.2.101/login.php into the search bar, and press enter. A warning page will appear, and in Google Chrome, "your connection is not private" will appear. In order to proceed to the site, click "Advanced" to the left of the "Back to Safety" button, and then click "Proceed anyway." The DinoBank customer portal will appear. The reason that the site is deemed unsafe is because the site certificate is invalid.

Figure 18

For the QueryTree application located at 10.0.2.103, simply open a browser, type 10.0.2.103/Account/Login into the search bar, and press enter. The QueryTree login page will be displayed, and next to the URL, “Not secure” will appear. Additionally, no site certificate is present.

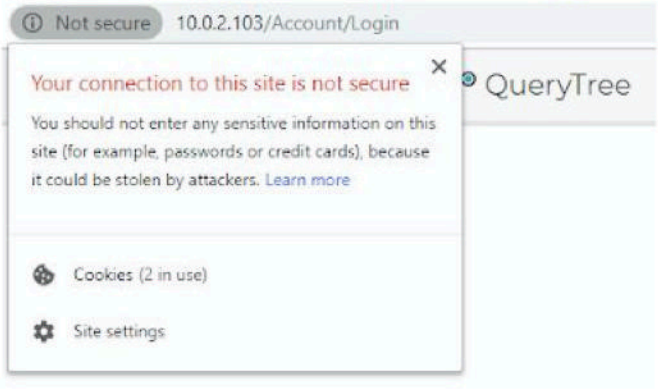


Figure 19

As a result, HTTPS will not work properly on these sites.

Impact:
Because https will not work properly on these sites, sensitive customer and company data could be intercepted by an attacker listening on the wire since client-server communication is not encrypted. Additionally, the invalid site certificate on the customer banking portal may cause customers to lose trust in DinoBank given that alert pages appear in all major browsers when https is improperly configured.

Recommendation:
In order to be properly configured to support an https connection, a valid site certificate must be present and installed, and https must be required server-side. All web servers must be upgraded to utilize TLS 1.2 with a certificate signed by a trusted root CA.

5.4.5 IVR Server Exposure (SID-405)

Vulnerability ID: SID-405
Vulnerability Type: Network Segmentation
Threat Level: Low
MITRE ATT&CK Mitigation: M1035 Limit Access to Resource Over Network

Description:
The Asterisk server was exposed on the DinoBank network, visible at 10.0.2.102:5038. This allows for Asterisk API calls to be made directly to the system.

Steps to reproduce:
On a machine with curl installed, execute the following command:

```
curl 10.0.2.102:5038
```

This opens an Asterisk Manager Interface prompt that allows for API calls to be made directly according to the official documentation.

Impact:
Incorrect login attempts did not appear to lead to account lockout. Using the `Action: login` API call, an attacker can potentially gain access to the administrator account via password guessing given that there is no limit to the number of attempts that can be made to log in.

Recommendation:
The Asterisk server should not be exposed on all network interfaces, and should be exposed only on the loopback interface.

5.4.6 Anonymous Share Enumeration (SID-406)

Vulnerability ID: SID-406
Vulnerability Type: Access Control
Threat Level: Low
MITRE ATT&CK Mitigation: M1015 Active Directory Configuration

Description:
On the branch domain controllers, incorrect access controls allowed for SMB shares to be enumerated from a null session. This allows attackers to gain insight into the structure of the network.

Steps to reproduce:

Using metasploit, use the `auxiliary/scanner/smb/smb_enumshares` module and set RHOSTS to 10.0.10.100, 10.0.11.100, or 10.0.12.100. Running the module shows that the domain controllers all have the SYSVOL and NETLOGON shares in addition to the standard C$, ADMIN$, and IPC$.

Impact:
As it stands, this is relatively low impact given that anonymous read access to these shares was not enabled and these are all default domain controller shares.

Recommendation:
Disable anonymous share enumeration on the branch domain controllers.

5.4.7 Ether web API source code disclosure (SID-407)

Vulnerability ID: SID-407 Vulnerability Type: Data Security Threat Level: Low MITRE ATT&CK Mitigation: M1021 Restrict Web-Based Content

Description:
On the HTML source of the ether application running on http://10.0.2.115:8000, there were links to the Ajax backend source code on the HTML source code.

Steps to reproduce:
Users can go to http://10.0.2.115:8000, right click, and go to inspect element. Then, they could click the linked .js files to view the backend source code.

Impact:
This would be a medium impact, as back-end source code disclosure may allow attackers to identify not-normally-seen vulnerabilities in the application. However, since the information gained from the source code disclosure didn't lead to any vulnerabilities, this has been reclassified as a low impact vulnerability.

Remediation:
Remove the relative URLS that link to the source code from the HTML document.

5.4.8 Usage of SMBv1 (SID-408)

Vulnerability ID: SID-408
Vulnerability Type: Software Updates
Threat Level: Low
MITRE ATT&CK Mitigation: M1051 Update Software

Description:
All Windows systems supported SMBv1 communications. This protocol has been outdated for a number of years, and Microsoft strongly recommends dropping support. Since DinoBank has no Windows XP or Windows Server 2003 systems, there is no reason to support SMBv1.

Steps to reproduce:
To verify this, in metasploit, use the "auxiliary/scanner/smb/smb1" module. Set the RHOSTS to any windows system IP, and run the module.

Impact:
SMBv1 is historically vulnerable to a number of exploits. The well known ransomware "WannaCry," for example, used a vulnerability in the SMBv1 protocol. This has since been patched, but it remains an inherently insecure protocol.

Remediation:
Disable SMBv1 support on all Windows systems.

5.4.8 Outdated Apache Server Version (SID-409)

Vulnerability ID: SID-409
Vulnerability Type: Software Updates
Threat Level: Low
MITRE ATT&CK Mitigation: M1051 Update Software

Description:
Network scans revealed a web server running an outdated version of Apache httpd on 10.0.1.33.

Steps to reproduce:
On a machine with nmap installed, run the following command:

```
nmap 10.0.1.33
```

This will reveal services available at this IP, one of which being Apache httpd v2.4.29 running on port 80.

Impact:
Due to this server running outdated software, it is potentially open to multiple attack vectors as detailed in the version's CVE details. These CVEs require very specific conditions to work properly, and thus this poses a low risk to DinoBank.

Remediation:
The server should be upgraded to the most recent stable version of Apache httpd, which is 2.4.41 at the time of this penetration test.

6. Non-Findings

6.1 Access Control

ID	Title	Description
NF-101	Hardened SSH Server	SSH servers were updated and requires public key authentication.
NF-102	Restrict RPC Null Session	The RPC service properly restricted null sessions.
NF-103	Lack of linux privilege escalation	Privilege escalation on the CORE-01 server was unsuccessful.

6.1.1 Hardened SSH Server (NF-101)

All linux servers used the latest version of SSH, and they all require public key authentication. This prevented us from gaining access to more linux servers, as we only had passwords for the SSH user.

6.1.2 Restrict RPC Null Session (NF-102)

All Windows systems restricted unauthenticated users from establishing a null session with the RPC service.

6.1.3 Lack of linux privilege escalation (NF-103)

We were unable to privilege escalate from the postgres user to the ubuntu or the root user on the CORE-01 server. This prevented us from gaining a fuller view of various configuration settings, such as redis configurations.

6.2 Input Validation

ID	Title	Description
NF-201	Lack of SQL injection	Web applications across Dinobank's network are written to prevent SQL injection attacks.

NF-202	Lack of XSS on Wiki and OnlineBanking	The Media Wiki and OnlineBanking applications prevented XSS

6.2.1 SQL injection input sanitization (NF-201)

The web servers on 10.0.1.205 and https://10.0.2.101 are both patched against SQL injection, preventing us from leaking database information.

6.2.2 Lack of XSS on Wiki (NF-202)

The MediaWiki at 10.0.1.31 allowed for editing of HTML pages. However, all attempts to embed JavaScript were properly sanitized. Similarly, the /alerts.php page reflected a GET parameter, but the HTML was properly sanitized to prevent javascript execution.

6.3 Network Access

ID	Title	Description
NF-301	MSSQL Listening on Loopback Adapter	The MSSQL database was listening on a loopback adapter and not exposed.

6.3.1 MSSQL Listening on Loopback Adapter (NF-301)

The MSSQL database on 10.0.1.50 was listening on a loopback adapter and therefore was not accessible on the open internet.

6.4 Credential Management

ID	Title	Description
NF-401	Password hashes resistant to cracking	All permitted attempts to crack password hashes failed
NF-402	SSH using private key authentication	All SSH servers were using private key authentication

6.4.1 Password hashes resistant to cracking (NF-401)

No passwords were recovered through hash cracking attempts. This potentially indicates that password policies are improved. Before attempting to crack any hashes, explicit permission was sought.

6.4.2 SSH using private key authentication (NF-402)

All SSH servers had switched from password authentication to SSH private key authentication. This makes gaining remote access significantly more difficult.

6.5 Data Security

ID	Title	Description
NF-501	MediaWiki was logged	MediaWiki server used IIS logging
NF-502	Centralized Windows logging	Splunk forwarders centralized logging

6.5.1 MediaWiki was logged (NF-501)

The MediaWiki server at 10.0.1.31 used default IIS logging to capture requests to the server, edits to webpages, and user authentication.

6.5.2 Centralized Windows logging (NF-502)

All Windows machines accessed had a Splunk forwarder installed that logged Powershell commands as well as Sysmon events and forwarded them to a centralized management server. Sysmon was configured with the SwiftOnSecurity Sysmon config, which is an effective file for finding common attack pathways.

6.6 Software Updates

ID	Title	Description
NF-601	Updated Linux Kernel	Linux kernel version on CORE-01 is up to date.

NF-602	Updated Windows Version	Windows OS is up to date and patched

6.6.1 Updated Linux Kernel (NF-601)

The linux kernel on CORE-01 is 5.0.0, which is the latest linux kernel version. This prevents users to privilege escalate to the root user through a kernel exploit.

6.6.2 Updated Windows Version (NF-602)

The Windows operating system, Windows Server 2016 Datacenter 14393, was up to date with the latest long term servicing branch and was fully patched.

7. Comparison to Previous Engagement

7.1 Overall Summary

Of the 20 SIDs reported previously, 7 were fixed and 13 remain the same or fixed insufficiently. None of the critical vulnerabilities were patched, however attempts were made to resolve the vulnerable PostgreSQL database. Therefore we rate the response **insufficient**: important issues were not fixed, and it would still be entirely possible to steal customer and employee PII or to illegally transfer money to an attacker's account.

Generally, the issues that were fixed were in custom DinoBank software, while the issues relating to infrastructure were not touched. The table below shows how many SIDs from each category there were and how many were fixed.

Vulnerability Category	Previous SID Count	Number Fixed
Critical	2	0
High	5	3
Medium	7	3
Low	6	1

7.2 Key Improvements Identified

7.2.1 Unauthenticated Account Creation

Description

Previously SID-102, this allowed arbitrary accounts to be created on the DinoBank customer portal. This could have been used to impersonate legitimate customers and steal their balances.

7.2.2 Croissant Bakery Removed

Description

Previously SID-105, this allowed unauthenticated users to edit index.php and deface the website.

7.2.3 SQL Injection, Account Number

Description

Previously SID-202, this allowed authenticated attackers to perform arbitrary SQL queries on 10.0.2.101. This could have been used to exfiltrate customer PII such as account numbers, bank balances, or account balances.

7.2.4 XSS to steal PHPSESSID cookies

Description

Previously SID-203, this allowed XSS attacks to be carried out on the website at 10.0.2.101. This could then be used to impersonate customers, log in to their accounts, and transfer money.

7.2.5 Password Policy

Description

Previously SID-402, this referred specifically to the credentials found in the database at 10.0.2.101. The password quality was drastically reduced, reducing the odds of attackers being able to brute force these customer and employee passwords.

However, during this engagement we were able to access many more credentials and asses the policies used to build them. Many local administrators in Windows workstations used a very, very easily guessable or crackable password. Furthermore, credentials used in other Linux services were based on dictionary words and would not be very difficult to crack.

7.2.6 Static API Keys

Description

Previously SID-402, this referred to static API keys used to access the 'bank' database API. Now, the credentials are loaded dynamically from a JSON file, which is an improvement.

7.2.7 Information Leak

Description

Previously SID-502, this referred to a process that leaked columns of the 'bank' database to unauthenticated users. This has now been patched.

7.3 Previous Key Findings Not Yet Remediated

7.3.1 PostgreSQL

Current ID: SID-101
Previous ID: SID-101
Criticality: Critical

7.3.2 FileZilla Anonymous FTP

Current ID: SID-204
Previous ID: SID-103
Criticality: High

7.3.3 Null SMB Sessions on Branch Domain Controller

Current ID: SID-302
Previous ID: SID-104
Criticality: Medium

7.3.4 FTP Bounce Attack

Current ID: SID-403
Previous ID: SID-108
Criticality: Low

7.3.5 Arbitrary Money Transfer

Current ID: SID-102
Previous ID: SID-201
Criticality: Critical

7.3.6 Exposed Domain Controller

Current ID: SID-207
Previous ID: SID-301
Criticality: High

7.3.7 Exposed PostgreSQL database

Current ID: SID-207
Previous ID: SID-302
Criticality: High

7.3.8 Exposed Remote Access Services

Current ID: SID-207
Previous ID: SID-303
Criticality: High

7.3.9 Plaintext Credentials and PII (PF-401)

Current ID: SID-101
Previous ID: SID-401
Criticality: Extreme

7.3.10 HTTP over HTTPS (PF-501)

Current ID: SID-404
Previous ID: SID-501
Criticality: Low

Description:
While this issue was technically fixed, the websites, such as my.dinobank.us, now have invalid certificates. Thus, when users visit the pages there is a message about the insecurity of the webpage. This will result in customers losing confidence in DinoBank and eventually hurt revenue.

7.3.12 Outdated Apache Web Server (PF-601)

Current ID: SID-409

Previous ID: SID-601
Criticality: Low

Description:
The Apache web server located at 10.0.1.33 was still running Apache httpd v2.4.29, which should be upgraded to the most recent version. At the time of this penetration test, this most recent version is 2.4.41.

7.3.13 Outdated SMB (PF-602)

Current ID: SID-408
Previous ID: SID-602
Criticality: Low

8. Special Assessment Areas

8.1 CroissantCoin Blockchain-based banking solution

The smart contract for the Croissant Coin should use the latest version of solidity. It was written in version 4.18, and the current version is 5.12. The use of the "now" keyword may now impose a small vulnerability in that miners can affect this to some degree since it refers to the timestamp on the block rather than the current time. After consideration is that while extraordinarily unlikely, due to the memory layout of smart contracts, two users' balanced and allowances may be stored at the same memory address. This is a result of the fact that solidity uses a form of hashing to determine where values are stored in memory, and any collision will result in the same balance or allowance to be used by multiple individuals. If dinobank is to scale to millions or even billions of users, some sort of advanced handling for this situation may need to be designed.

A far more serious concern lies in the possibility of a double spending approval attack. At times, a user may choose to change the allowance they are providing to another user. If the other user is closely monitoring the blockchain and notices this change before it is processed, they may spend their allowance. Since each block processes multiple transactions, this would trigger a race condition possibly allowing the user to spend their allowance before the change and again after the change. The ERC20 standard says to fix this through the client, but an attacker may design their own client to bypass

this. The correct client fix involves listening for approval and transfer events and triggering callbacks which run the transaction, which cannot be intercepted by attackers.

8.2 Interactive Voice Response (IVR) System

Dinobank provided a phone number for use in testing involving their Interactive Voice Response (IVR) system for Online Banking. User authentication vulnerabilities were explored first. The service required both a 9 digit tax ID in conjunction with a 4 digit PIN, so authentication via brute-force was not an option. Furthermore, the vague error message upon unsuccessful authentication was vague, not specifying which credential (tax ID or password) was incorrect. This is a positive security measure that deters brute forcing. Thus, user authentication vulnerabilities are **nonfindings.**

Next, the menu listing upon authenticating into the IVR system was explored. Hidden features not mentioned by the menu were investigated, however all invalid menu choices resulted in call disconnection or the system listing a pre-recorded generic error message. Extended DTMF tones were also utilized to look for hidden features, to no avail. This is a **nonfinding.**

Next, malformed inputs were used in an attempt to fuzz the IVR system. Special characters were utilized in an SQL injection-like attack, but the system was resistant (as described by Rahul Sasi, Black Hat Security Conference: Europe 2012, *IVR Security*). However, extremely long inputs caused the system to hang. This could allow attackers to send multiple lengthy inputs to the IVR system, which could eventually result in a Denial of Service. This is a **low finding**, and was not attempted to maintain system uptime.

Finally, the Asterisk Management API was discovered running on the online banking server. All action API calls were unavailable to unauthenticated users, which is a **nonfinding**, but since there were no apparent lockouts enabled on the application, an attacker with sufficient time may successfully guess the password. To mitigate this issue, this application should not be exposed on all network interfaces, and should be exposed only on the loopback interface. This is a **low finding**.

8.3 Automated Teller Machine (ATM)

DinoBank provided us with a Tranax Hyosung Mini Bank 1500 ATM. We began our assessment by finding the manual online in order to gain information about potential default credentials. We attempted to enter "Operator Mode" which is roughly equivalent to administrator mode. We were unable to enter this mode using the default password, suggesting it had been changed by dinobank. This is good news as it prevents easy access by anyone with access to a manual. This is a **nonfinding**.

Next, we explored our own account. We were provided with an ATM card from DinoBank that had 0 funds in it but allowed us to interact with the customer portal. When we first used the card we were prompted to set up a pin. I set the pin as 0000 for testing. I then attempted to login to my account with an incorrect pin and found that I was able to successfully authenticate with a handful of pins that were four or more digits. This leads us to believe that anyone who managed to get access to an atm card would be able to access the funds and authenticate with a random pin. This is a **critical finding**.

Finally, at one point an ATM tech came to do an audit on our machine. He entered operator mode by typing the password in plain view and then left it in operator mode when he exited. This gave us two things: our initial access to the machine in operator mode as well as allowed us to see the password and write it down for future use. Although this should not have happened the access did not allow us to do much other than see statistics of usage. There is a master password for any really critical functionalities, which we did not have access to. Thus this is a **low finding**.

8.4 Active Directory Environment

After gaining access to DinoBank's Active Directory Environment, we performed an extensive audit that examined stale objects, privileged accounts, trusts, and group policies. Through our analysis, we assess that the company's Domain Risk Level is Severe. As shown in the below graph and described above, there were many high and critical configuration issues that weakened the company's system. Additionally, DinoBank is primarily a Windows-based organization which means this has a high impact to the company if compromised.

Risk model

Stale Objects	Privileged accounts	Trusts	Anomalies
Inactive user or computer	Account take over	Old trust protocol	Backup
Network topography	ACL Check	SID Filtering	Certificate take over
Object configuration	Admin control	SIDHistory	Golden ticket
Obsolete OS	Irreversible change	Trust impermeability	Local group vulnerability
Old authentication protocols	Privilege control	Trust inactive	Network sniffing
Provisioning			Pass-the-credential
Replication			Password retrieval
Vulnerability management			Reconnaissance
			Temporary admins
			Weak password

Legend:
- score is 0 - no risk identified but some improvements detected
- score between 1 and 10 - a few actions have been identified
- score between 10 and 30 - rules should be looked with attention
- score higher than 30 - major risks identified

Figure 20

Stale Objects (Medium)

Primarily, this audit category had a number of smaller issues that would enable attackers to collect significant information on the domain and generally lowered the security of the Windows systems.

Domain Computer Registration Process

Description:

This setting ensures that basic users cannot register extra computers in the domain. By default, a basic user can register 10 computers within the domain when only Administrators should need this privilege.

Recommendation:

To solve the issue limit the number of extra computers that can be registered by a basic user. It can be reduced by modifying the value of ms-DS-MachineAccountQuota to zero (0). Another solution can be to remove altogether the authenticated users group in the domain controllers policy.

SMBv1 is Enabled (Low)

Description:

All 4 Domain Controllers in the company network had SMB v1 enabled. An SMB downgrade attack can be used to obtain credentials or execute commands on behalf of a user by using the SMB v1 protocol.

Recommendation:

Microsoft highly recommends to disable SMB v1 whenever it is possible on both client and server side. In addition, since all of the Windows machines we located in the environment are running Windows 10 and Windows Server 2016, there are not any legacy reasons for keeping this setting enabled.

Privileged Accounts (Critical)

More than 95% of AD Administrative Accounts are inactive

Description:

We determined that more than 95% of the Active Directory Administrative Accounts (users with Domain Admin Privileges) had no login within the last 6 months. By examining the attributes stored on each account, we can collect this information. It is important to ensure that Administrator Accounts are actively used to follow the least privileged access model.

Recommendation:

If users no longer have a reason to utilize their AD Admin account, their privileges should be revoked. Another alternative that may have less impact on operations would be to temporarily disable these accounts while they are not in use. In addition, we also suggest audited all of the Domain Admins to ensure each require this level of access for their job. Because these accounts have full control of every Windows machine in the network, it is absolutely essentially that Domain Admin accounts are properly secured.

Presence of Admin accounts which lack the flag "this account is sensitive and cannot be delegated"

Description:

As stated previously, Active Directory Administrative Accounts represent a ripe target for attackers given their authority to access any Windows computer. Domain Admin accounts should all have the flag "This account is sensitive and cannot be delegated" enabled which limits the risk of credential theft.

Recommendation:

All accounts in the Domain Admins group should have the "This account is sensitive and cannot be delegated" flag. In addition, we recommend that company users have a separate user account for all of their administrative actions that is separate from their normal account.

Ensure that dangerous settings & privileges are not granted to everyone by GPO

Local Policies/User Rights Assignment

Policy	Setting
Act as part of the operating system	Everyone
Allow log on through Terminal Services	Domain Users

Figure 21

Description:

The purpose of this setting is to ensure that standard users are not granted dangerous privileges. We identified that users were granted the **SeTcbPrivilege** privilege through the SUPAR SECUR group policy. SeTcbPrivilege is the privilege used to **"Act on behalf of the operating system"**. This is a privilege normally reserved to the SYSTEM user. This procedure allow any users to act as SYSTEM. Additionally, this group policy stored passwords in reverse encryption.

Recommendation:

Modify the SUPAR SECUR group policy to not grant this privilege to Everyone. In addition, the “Store passwords using reversible encryption” should be set to disabled.

Trusts (Non-Finding)

There were no Domain or Forest Trusts identified between the domains which means there were no Forest-to-Forest Trust relationships or similar issues that could be abused.

Anomalies (High)

All local Administrator accounts have the same password

Description:
As noted above in the Findings section, all of the local Administrator accounts on the workstations and servers had the same password. As a result, if an attacker gains access to one workstation and obtains the admin hash or password, they can use this to access every other machine. The Administrator accounts were all enabled across the environment.

Recommendation:
We strongly recommend installing Microsoft's Local Administrator Password Solution (LAPS) tool. This software can be applied through group policy, and it randomizes the local administrative password on every Windows computer. Based on the fact that DinoBank does not appear to have a provisioning process or password vault solution to manage these accounts, this would be an effective way to make it difficult for an attacker to pivot through the environment.

Weak account policy requirements

Password policies

Note: PSO (Password Settings Objects) will be visible only if the user which collected the information has the permission to view it.
PSO shown in the report will be prefixed by "PSO:"

Policy Name	Complexity	Max Password Age	Min Password Age	Min Password Length	Password History	Reversible Encryption	Lockout Duration	Reset account counter locker after
Default Domain Policy	False	60 day(s)	0 day	6	0	False	5 minute(s)	5 minute(s)
SUPAR SECUR	False	Not Set	Not Set	4	Not Set	True	1 minute(s)	1 minute(s)

Figure 22

Description:

Both the Default Domain Policy and SUPAR SECUR group policies contained a number of weak settings including 6 and 4 character min password lengths respectively, reversible encryption, no enforcement of complexity requirements, short lockout durations, and other settings such as “Let Everyone permissions apply to anonymous.” These settings encourage and allow users to configure bad settings that are **not** in compliance with your financial regulations that your organization is under.

Recommendation:

We recommend modifying both of these group policies to follow a published secure host baseline/STIG like those published by NIST, the Department of Defense, or the National Security Agency. For example, passwords should be a minimum of 9 characters, and all passwords should be required to meet complexity requirements.

9. Conclusion

The network was given a critical severity due to the large volume of sensitive customer data that is exposed across the network, poor storage of PII, and poor credential policies that allowed for exploitation of the entire Windows network. The main vulnerabilities discovered include:

- Lack of authorization to services such as FTP and the banking web application
- Poor network segmentation leaving critical services such as FTP, PostgreSQL, and RDP exposed
- Numerous employee passwords left in plaintext across the network.
- Sensitive customer data left unencrypted in plaintext.
- Easy-to-guess passwords on administrator-level accounts across the network.

In response to the findings and vulnerabilities listed above, we make the following recommendations to DinoBank:

- Perform network segmentation to hide exposed services such as FTP, PostgreSQL, and RDP.
- Implement proper authentication to prevent anonymous users from accessing application and performing arbitrary fund transfers.
- Hash employee and customer passwords using a hashing algorithm such as SHA3-256. Also implement new password policy to improve password strength.
- Encrypt sensitive customer data, such as social security numbers, to protect customer privacy.

Following these steps will improve DinoBank's performance in the audit categories relating to password policy and banking core applications. However, in order to improve performance in the security governance and management categories, DinoBank additionally needs to:

- Encourage trust and communication between security leadership and employees either through teamwork education programs or finding new leadership
- Invest in employee training regarding secure development and password protection practices.