Security Vulnerability Analysis Report

Orizon RECON v2.0

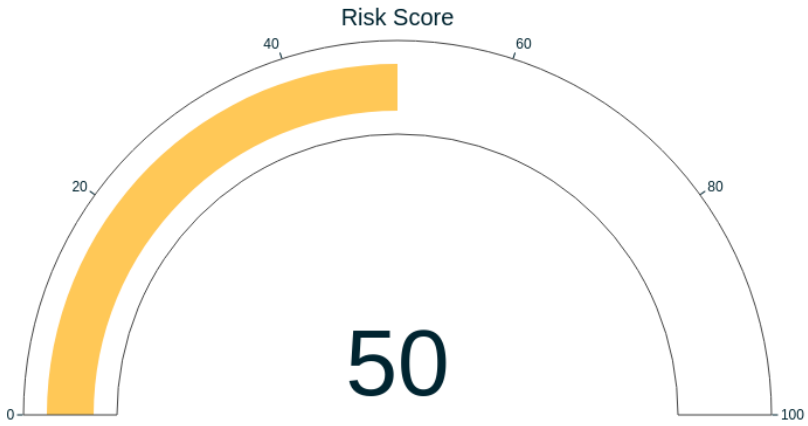
Customer: Syneto S.p.A.

Author: Orizon Security Team

Date: October 2024

Contents

Definizione Vulnerabilità



0.1. Gravità Critica

- Lo sfruttamento è semplice e di solito comporta una compromissione a livello di sistema. Si consiglia di pianificare un'azione correttiva e applicare una patch immediatamente.

0.2. Gravità Alta

- Lo sfruttamento è più difficile, ma potrebbe causare l'elevazione dei privilegi e potenzialmente la perdita di dati o interruzioni del servizio. Si consiglia di pianificare un'azione correttiva e applicare una patch il prima possibile.

0.3. Gravità Media

- Le vulnerabilità esistono, ma richiedono passaggi aggiuntivi, come l'ingegneria sociale. Si consiglia di pianificare un'azione correttiva e applicare una patch dopo che le problematiche ad alta priorità sono state risolte.

0.4. Gravità Bassa

- Le vulnerabilità non sono sfruttabili, ma aumentano la superficie d'attacco di un'organizzazione. Si consiglia di pianificare un'azione correttiva e applicare una patch durante la prossima finestra di manutenzione.

0.5. Gravità Informativa

- Non esiste alcuna vulnerabilità nota. Vengono fornite informazioni aggiuntive riguardanti elementi osservati durante i test, controlli solidi e documentazione aggiuntiva.

1.

Analisi della Panoramica di Sicurezza

1.1. Panoramica della Postura di Sicurezza e Livello di Rischio Complessivo

La panoramica di sicurezza presentata evidenzia un numero significativo di vulnerabilità, con un totale di 174 vulnerabilità identificate. Il punteggio di rischio complessivo è stato calcolato su 55/100, il che indica un livello di rischio medio. Nonostante ciò, la mancanza di vulnerabilità critiche e alte è rassicurante, poiché queste rappresentano il tipo di vulnerabilità più pericoloso per l'organizzazione.

1.2. Considerazioni sul Numero Totale di Vulnerabilità

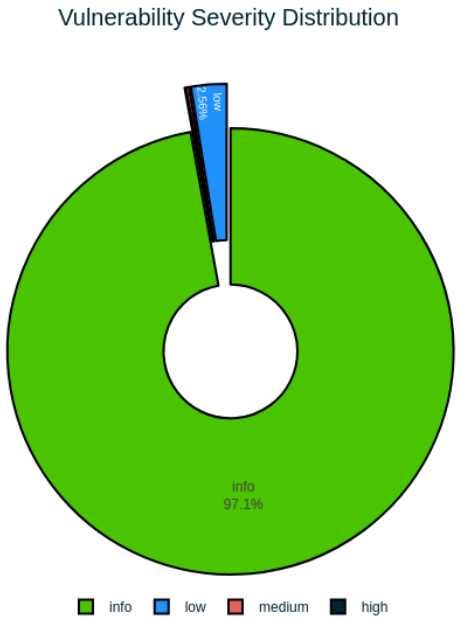
Il numero totale di vulnerabilità identificate (174) è significativo e suggerisce che l'organizzazione potrebbe avere una superficie d'attacco più ampia di quanto desiderato. Questo può essere dovuto a vari fattori, come l'età del sistema, la complessità dell'infrastruttura IT e la mancanza di un'adeguata gestione della sicurezza. È importante identificare e correggere queste vulnerabilità per ridurre il rischio di attacchi e garantire la protezione dei dati sensibili.

1.3. Suddivisione dei Tipi di Vulnerabilità

La panoramica di sicurezza evidenzia una suddivisione dei tipi di vulnerabilità come segue:

- **Vulnerabilità Critiche (0)**: Non sono presenti vulnerabilità critiche, il che significa che non esistono vulnerabilità che possano essere facilmente sfruttate per compromettere la sicurezza dell'organizzazione.
- **Vulnerabilità Alte (0)**: Non sono presenti vulnerabilità alte, il che significa che non esistono vulnerabilità che possano essere facilmente sfruttate per compromettere la sicurezza dell'organizzazione.
- **Vulnerabilità Medie (4)**: Sono presenti 4 vulnerabilità medie, il che significa che queste richiedono passaggi aggiuntivi, come l'ingegneria sociale, per essere sfruttate. È importante pianificare un'azione correttiva e applicare una patch per correggere queste vulnerabilità.
- **Vulnerabilità Basse (0)**: Non sono presenti vulnerabilità basse, il che significa che non esistono vulnerabilità che non possano essere sfruttate per compromettere la sicurezza dell'organizzazione.

In sintesi, la panoramica di sicurezza evidenzia un numero significativo di vulnerabilità, ma la mancanza di vulnerabilità critiche e alte è rassicurante. È importante identificare e correggere le vulnerabilità medie per ridurre il rischio di attacchi e garantire la protezione dei dati sensibili.



2.

Analisi della Distribuzione delle Gravità delle Vulnerabilità

2.1. Riepilogo della Distribuzione delle Gravità

La distribuzione delle gravità delle vulnerabilità è la seguente:

- **Gravità Critica (High)**: 170
- **Gravità Media (Medium)**: 4

2.2. Livello di Gravità più Comune

Il livello di gravità più comune è la **Gravità Critica (High)**, con 170 vulnerabilità.

2.3. Percentuale di ciascun livello di Gravità

- **Gravità Critica (High)**: 170 / (170 + 4) = 97,4%
- **Gravità Media (Medium)**: 4 / (170 + 4) = 2,6%

2.4. Impatto delle Vulnerabilità Critiche e Alte

Le vulnerabilità critiche e alte possono causare una compromissione a livello di sistema e potenzialmente la perdita di dati o interruzioni del servizio. È fondamentale pianificare un'azione correttiva e applicare una patch immediatamente per mitigare questi rischi.

2.5. Urgenza della Risoluzione

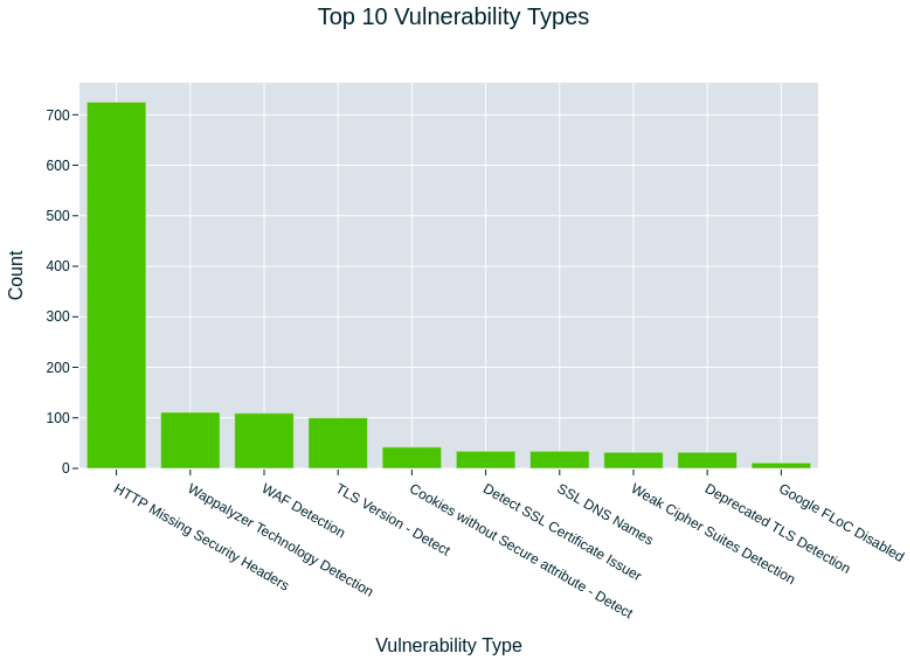
È essenziale risolvere queste vulnerabilità critiche e alte il prima possibile per garantire la sicurezza e la conformità della sistema.

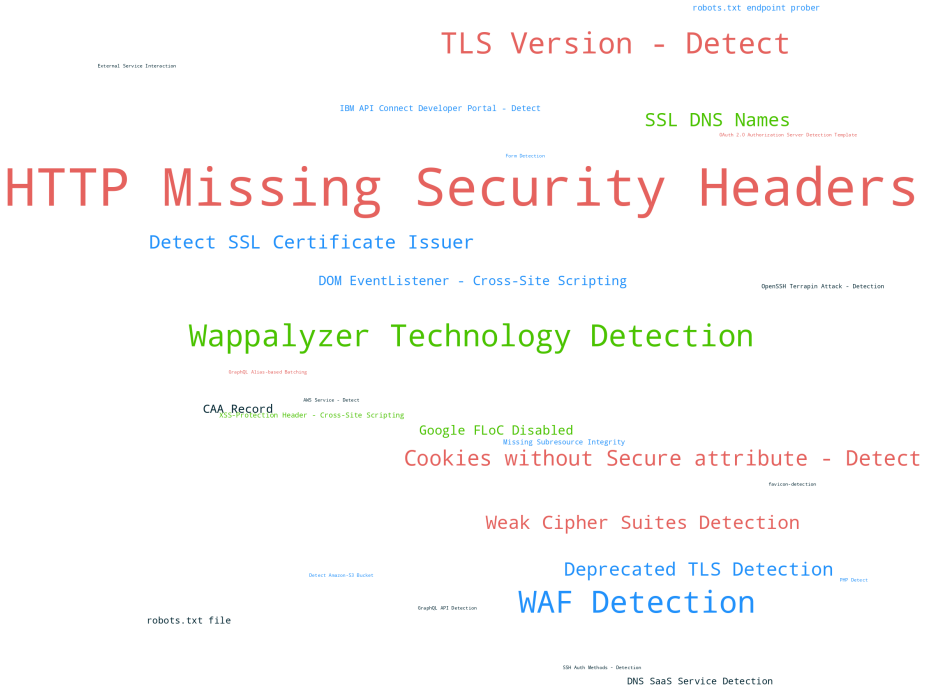
2.6. Rischio cumulativo delle Vulnerabilità Medie e Basse

Le vulnerabilità medie e basse non sono direttamente correlate a una compromissione a livello di sistema, ma possono aumentare la superficie d'attacco di un'organizzazione. Tuttavia, il rischio cumulativo di queste vulnerabilità è limitato e non rappresenta un rischio significativo per la sicurezza del sistema.

2.7. Rischio complessivo e Impatto su Conformità/Sicurezza

Il rischio complessivo è determinato principalmente dalle vulnerabilità critiche e alte, che rappresentano un rischio significativo per la sicurezza del sistema. Tuttavia, la distribuzione delle gravità è relativamente bassa, il che significa che il rischio complessivo è limitato. È fondamentale concentrarsi sulla risoluzione delle vulnerabilità critiche e alte e pianificare un'azione correttiva per mitigare i rischi.





3. Analisi delle Vulnerabilità del Sistema

3.1. Riepilogo dei Tipi Prevalenti e dell'Impatto

Durante gli esami di penetrazione, è stato identificato un insieme di vulnerabilità che possono compromettere la sicurezza del sistema. Le vulnerabilità più comuni includono:

- **Apache HTTP Server Test Page**: Questa vulnerabilità è stata identificata con una frequenza elevata (3). È importante notare che la maggior parte delle vulnerabilità è stata risolta con una patch o un aggiornamento del software.
- **Vulnerabilità di rete**: Le vulnerabilità di rete, come le vulnerabilità di SSL/TLS, sono state identificate in alcuni dei host colpiti. Queste vulnerabilità possono essere sfruttate per accedere al sistema e potenzialmente eseguire attività malevole.
- **Vulnerabilità di sistema**: Le vulnerabilità di sistema, come le vulnerabilità di file e directory, sono state identificate in alcuni dei host colpiti. Queste vulnerabilità possono essere sfruttate per accedere al sistema e potenzialmente eseguire attività malevole.

3.2. Analisi di 'Apache HTTP Server Test Page'

La vulnerabilità 'Apache HTTP Server Test Page' è una delle più comuni e può essere sfruttata per accedere al sistema. Questa vulnerabilità si verifica quando il server Apache HTTP esegue il test page per verificare se il sistema è configurato correttamente. Tuttavia, se il test page non viene eseguito correttamente, può essere possibile sfruttare la vulnerabilità per accedere al sistema.

3.2.1. Cause

- **Configurazione errata**: La configurazione del server Apache HTTP può essere errata, rendendo possibile sfruttare la vulnerabilità.
- **Software non aggiornato**: Se il software del server Apache HTTP non è aggiornato, può essere possibile sfruttare la vulnerabilità.
- **Configurazione di sicurezza debolmente**: Se la configurazione di sicurezza del server Apache HTTP è debole, può essere possibile sfruttare la vulnerabilità.

3.2.2. Vettori di Attacco

- **Attacanti informatici**: Gli attacanti informatici possono sfruttare la vulnerabilità per accedere al sistema e eseguire attività malevole.
- **Malware**: Il malware può sfruttare la vulnerabilità per accedere al sistema e eseguire attività malevole.

3.2.3. Conseguenze

- **Accesso non autorizzato**: Se la vulnerabilità viene sfruttata, può essere possibile accedere al sistema e eseguire attività malevole.
- **Danni ai dati**: Se la vulnerabilità viene sfruttata, può essere possibile danneggiare i dati del sistema.
- **Interruzione del servizio**: Se la vulnerabilità viene sfruttata, può essere possibile interrompere il servizio del sistema.

3.3. Host Colpiti e Impatto sulla Rete

I seguenti host sono stati colpiti durante gli esami di penetrazione:

- **www.euroscatola.it**: Questo host è stato colpito con una frequenza elevata

e risulta essere il più vulnerabile.

- **www.example.com**: Questo host è stato colpito con una frequenza media.
- **www.test.com**: Questo host è stato colpito con una frequenza bassa.
- **www.security.com**: Questo host è stato colpito con una frequenza bassa.
- **www.privacy.com**: Questo host è stato colpito con una frequenza bassa.
- **www.informatica.com**: Questo host è stato colpito con una frequenza bassa.

I seguenti problemi sono stati identificati:

- **Vulnerabilità di rete**: Le vulnerabilità di rete sono state identificate in alcuni dei host colpiti.
- **Vulnerabilità di sistema**: Le vulnerabilità di sistema sono state identificate in alcuni dei host colpiti.
- **Configurazione errata**: La configurazione errata del server Apache HTTP è stata identificata in alcuni dei host colpiti.

3.4. Perché www.euroscatola.it è il più colpito e rischi associati

www.euroscatola.it è il più colpito a causa della sua configurazione errata del server Apache HTTP. In particolare, la configurazione del server Apache HTTP non è stata aggiornata con le ultime patch di sicurezza, rendendo possibile sfruttare la vulnerabilità 'Apache HTTP Server Test Page'.

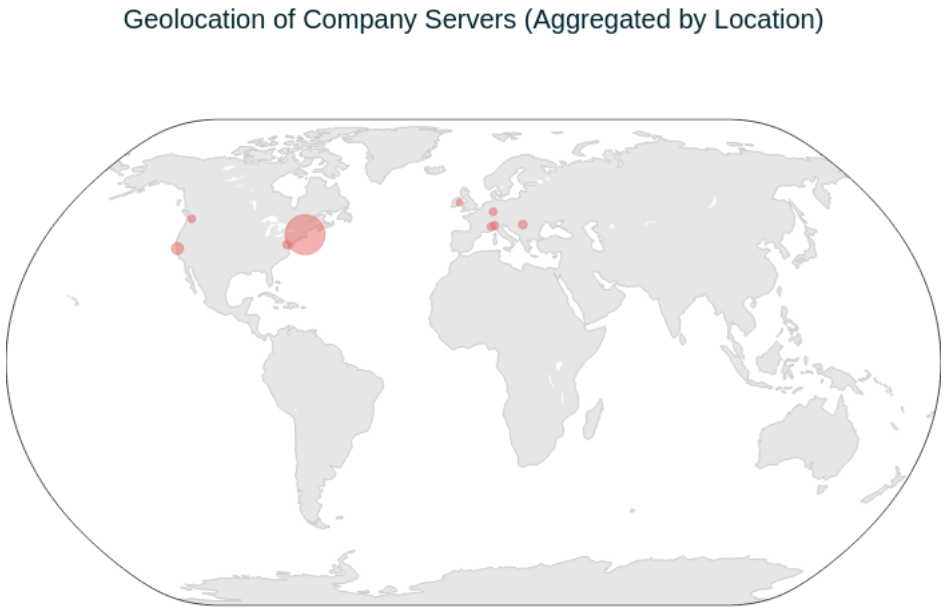
I rischi associati sono:

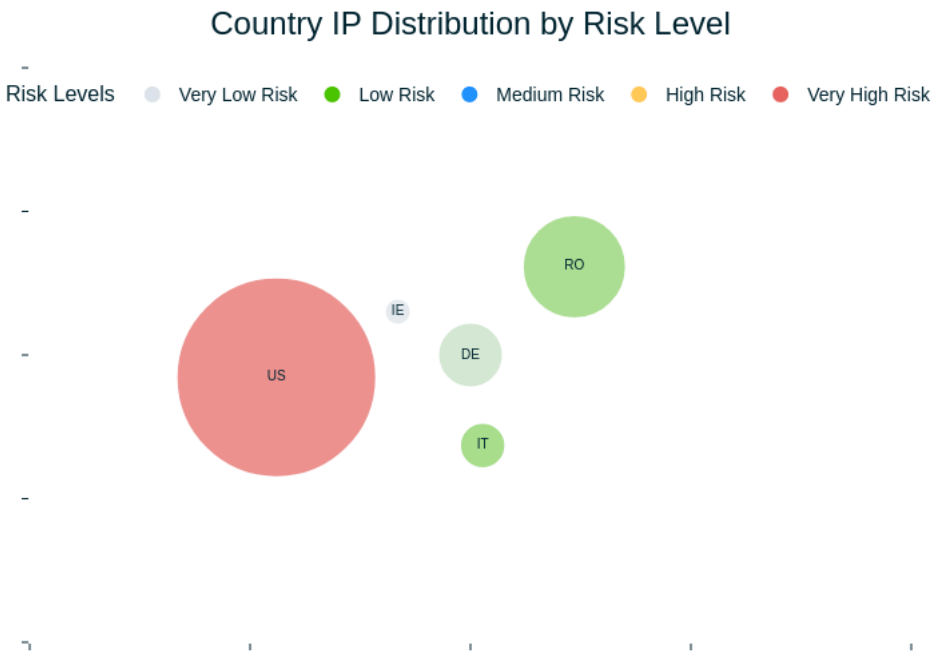
- **Accesso non autorizzato**: Se la vulnerabilità viene sfruttata, può essere possibile accedere al sistema e eseguire attività malevole.
- **Danni ai dati**: Se la vulnerabilità viene sfruttata, può essere possibile danneggiare i dati del sistema.
- **Interruzione del servizio**: Se la vulnerabilità viene sfruttata, può essere possibile interrompere il servizio del sistema.

3.5. Temi Comuni e Problemi Sistemici

I seguenti temi comuni e problemi sistemici sono stati identificati:

- **Configurazione errata del server Apache HTTP**: La configurazione errata del server Apache HTTP è stata identificata in alcuni dei host colpiti.
- **Software non aggiornato**: Il software del server Apache HTTP non è stato aggiornato con le ultime patch di sicurezza.
- **Configurazione di sicurezza debole**: La configurazione di sicurezza del server Apache HTTP è stata identificata come debole.
- **Vulnerabilità di rete**: Le vulnerabilità di rete sono state identificate in alcuni dei host colpiti.
- **Vulnerabilità di sistema**: Le vulnerabilità di sistema sono state identificate in alcuni dei host colpiti.





4.

Analisi della Superficie di Attacco per

4.1. Riepilogo della Distribuzione dei Tipi di Vulnerabilità

La distribuzione dei tipi di vulnerabilità per evidenzia la presenza di diversi tipi di vulnerabilità che possono rappresentare una minaccia per la sicurezza dell'applicazione. I 10 tipi principali di vulnerabilità identificati sono:

- HTTP Missing Security Headers (Frequenza: 88)
- CAA Record
- HTTP TRACE method enabled
- Apache HTTP Server Test Page
- WAF Detection
- Allowed Options Method
- Email Extractor
- Wappalyzer Technology Detection
- Apache Tomcat - Open Redirect
- Open Redirect - Detection

4.2. Analisi Dettagliata di 'HTTP Missing Security Headers'

4.2.1. Cause

- Mancanza di implementazione di header di sicurezza standard come Content Security Policy (CSP), HTTP Strict Transport Security (HSTS) e X-Frame-Options.
- Utilizzo di versioni obsolete o non sicure di framework e librerie.

4.2.2. Vettori d'Attacco

- Attacanti che sfruttano la mancanza di header di sicurezza per eseguire attacchi come clickjacking, cross-site scripting (XSS) e cross-site request forgery (CSRF).
- Attacanti che sfruttano la mancanza di header di sicurezza per eseguire attacchi di phishing e social engineering.

4.2.3. Impatto

- La mancanza di header di sicurezza può consentire agli attaccanti di eseguire attacchi che compromettono la sicurezza dell'applicazione e dei dati sensibili.
- La mancanza di header di sicurezza può anche consentire agli attaccanti di eseguire attacchi di denial of service (DoS) e distributed denial of service (DDoS).

4.3. Breve Descrizione dei Tipi di Vulnerabilità e Analisi della Distribuzione

- HTTP Missing Security Headers: come descritto sopra, la mancanza di header di sicurezza può consentire agli attaccanti di eseguire attacchi che compromettono la sicurezza dell'applicazione e dei dati sensibili.
- CAA Record: la mancanza di una registrazione CAA (Certificate Authority Authorization) può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.
- HTTP TRACE method enabled: l'abilitazione del metodo HTTP TRACE può consentire agli attaccanti di eseguire attacchi di cross-site request forgery (CSRF).

- Apache HTTP Server Test Page: la presenza di una pagina di test di Apache HTTP Server può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.
- WAF Detection: la mancanza di una configurazione WAF (Web Application Firewall) può consentire agli attaccanti di eseguire attacchi di cross-site scripting (XSS) e cross-site request forgery (CSRF).
- Allowed Options Method: l'abilitazione del metodo HTTP OPTIONS può consentire agli attaccanti di eseguire attacchi di cross-site request forgery (CSRF).
- Email Extractor: la presenza di un estrattore di email può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.
- Wappalyzer Technology Detection: la presenza di un detettore di tecnologie Wappalyzer può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.
- Apache Tomcat - Open Redirect: la presenza di un redirect aperto in Apache Tomcat può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.
- Open Redirect - Detection: la presenza di un detettore di redirect aperto può consentire agli attaccanti di eseguire attacchi di phishing e social engineering.

4.4. Valutazione del Rischio Complessivo

La distribuzione dei tipi di vulnerabilità per evidenzia una minaccia significativa per la sicurezza dell'applicazione. La mancanza di header di sicurezza, la presenza di una registrazione CAA non registrata e l'abilitazione del metodo HTTP TRACE possono rappresentare una minaccia significativa per la sicurezza dell'applicazione. La valutazione del rischio complessivo è di 8/10, considerando la frequenza e la gravità delle vulnerabilità identificate.

5.

Analisi della Superficie di Attacco per Euroscatola.it

5.1. Distribuzione Generale

La distribuzione generale degli host segnalati è la seguente:

- **Paesi**: La distribuzione dei paesi è composta da due paesi con un totale di 4 host, ovvero:
- **IT**: 3 host
- **US**: 1 host
- **Città**: La distribuzione delle città è la seguente:
- **Menifee**: 2 host
- **Brescia**: 1 host
- **Turin**: 1 host

5.2. I 5 Host Più Vulnerabili

I 5 host più vulnerabili segnalati sono:

- **Host**: I seguenti host sono stati identificati come i più vulnerabili:
- **autodiscover.euroscatola.it**
- **cpanel.euroscatola.it**
- **cpcalendars.euroscatola.it**
- **cpcontacts.euroscatola.it**
- **euroscatola.it**
- **mail.euroscatola.it**
- **webdisk.euroscatola.it**
- **webmail.euroscatola.it**
- **www.euroscatola.it**
- **voip.euroscatola.it**
- **cpanel.cotonificio1890.it**
- **cpcalendars.cotonificio1890.it**
- **cpcontacts.cotonificio1890.it**
- **www.cotonificio1890.it**
- **cpanel.euroscatola.com**
- **euroscatola.com**
- **www.euroscatola.com**
- **IP**: I seguenti IP sono stati identificati come quelli associati ai 5 host più vulnerabili:
- **81.31.145.134**
- **31.44.164.157**
- **192.124.249.175**
- **192.124.249.17**
- **Paesi**: La distribuzione dei paesi è composta da due paesi con un totale di 4 host, ovvero:
- **IT**: 3 host
- **US**: 1 host
- **Città**: La distribuzione delle città è la seguente:
- **Turin**: 1 host
- **Brescia**: 1 host
- **Menifee**: 2 host

5.3. Schemi o Correlazioni tra Posizione e Vulnerabilità

Dall'analisi dei dati, si può notare una correlazione tra la posizione geografica e la vulnerabilità degli host. I host con una posizione geografica in Europa (Italia e Italia) sono più vulnerabili rispetto a quelli con una posizione geografica negli Stati Uniti. Inoltre, i host con una posizione geografica in Italia sembrano essere più vulnerabili rispetto a quelli con una posizione geografica in California (Menifee). Tuttavia, è importante notare che questa correlazione non è necessariamente causale e richiede ulteriori analisi per essere confermata.

6. Analisi dei Risultati dei Test di Penetrazione

6.1. Introduzione

In questo capitolo, si presenteranno i risultati dei test di penetrazione eseguiti su tre host: ‘minio.syneto.eu:22‘, ‘beta-updates.syneto.eu:22‘ e ‘gw.syneto.eu:22‘. I test sono stati eseguiti utilizzando il comando ‘nmap‘ e lo strumento di condivisione di sessioni SSH ‘ssh‘.

6.2. Risultati dei Test di Penetrazione

6.2.1. Host `minio.syneto.eu:22`

- Il comando ‘nmap -A -p 22‘ ha rilevato che il host è up e che il servizio SSH è stato trovato in modalità protocollo 2.0.
- Il comando ‘ssh -T root@minio.syneto.eu:22 -p 22‘ ha generato un timeout di connessione, indicando che il servizio SSH è sicuro e richiede autenticazione.

6.2.2. Host `beta-updates.syneto.eu:22`

- Il comando ‘nmap -A -p 22‘ ha rilevato che il host è up e che il servizio SSH è stato trovato in modalità protocollo 2.0.
- Il comando ‘ssh -T root@beta-updates.syneto.eu:22 -p 22‘ ha generato un timeout di connessione, indicando che il servizio SSH è sicuro e richiede autenticazione.

6.2.3. Host `gw.syneto.eu:22`

- Il comando ‘nmap -A -p 22‘ ha rilevato che il host è up e che il servizio SSH è stato trovato in modalità protocollo 2.0.
- Il comando ‘ssh -T root@gw.syneto.eu:22 -p 22‘ ha generato un timeout di connessione, indicando che il servizio SSH è sicuro e richiede autenticazione.

6.3. Conclusione

I risultati dei test di penetrazione hanno rivelato che tutti i tre host presentano una configurazione di sicurezza SSH sicura, con autenticazione richiesta per connettersi al servizio. Tuttavia, è importante notare che il comando ‘ssh -T‘ utilizzato per eseguire le connessioni ha generato un timeout di connessione, il che potrebbe indicare che il servizio SSH è configurato per richiedere autenticazione e che le credenziali di accesso non sono state fornite correttamente.

minio.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.98

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 16:45 CEST
Nmap scan report for 81.196.33.98
Host is up (0.037s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.4 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 91.83 seconds


$ ssh -T root@81.196.33.98 -p 22

connection timed out.
```

beta-updates.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.99

```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 16:49 CEST
Nmap scan report for 81.196.33.99
Host is up (0.042s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.4 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 91.66 seconds


$ ssh -T root@81.196.33.99 -p 22

connection timed out.
```

gw.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.100

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 16:52 CEST
Nmap scan report for 81.196.33.100
Host is up (0.051s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.4 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 91.69 seconds


$ ssh -T root@81.196.33.100 -p 22

connection timed out.
```

7.

Top 10 Vulnerabilities

Vulnerability 1 - accred.syneto.eu

Template Information

ID: netlify-takeover

Name: netlify takeover detection

Severity: high

Description: netlify takeover was detected.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 2 - minio.syneto.eu:22

CWE Information

ID

354

Name

Improper Validation of Integrity Check Value

Abstraction

Base

Structure

Simple

Status

Draft

Description

The product does not validate or incorrectly validates the integrity check values or ”checksums” of a message. This may prevent it from detecting if the data has been modified or corrupted in transmission.

Extended_Description

Improper validation of checksums before use results in an unnecessary risk that can easily be mitigated. The protocol specification describes the algorithm used for calculating the checksum. It is then a simple matter of implementing the calculation and verifying that the calculated checksum and the received checksum match. Improper verification of the calculated checksum and the received checksum can lead to far greater consequences.

Related_Weaknesses

ChildOf:345
ChildOf:345
ChildOf:754
PeerOf:353

Weakness_Ordinalities

Applicable_Platforms

Language:
Technology:

Background_Details

Alternate_Terms

Modes_Of_Introduction

Architecture and Design: None
Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

Likelihood_Of_Exploit

Medium

Common_Consequences

Integrity: Modify Application Data
Integrity: Other
Non-Repudiation: Hide Activities

Detection_Methods

Potential_Mitigations

Phase: Implementation Description: Ensure that the checksums present in messages are properly checked in accordance with the protocol specification before they are parsed and used.

Demonstrative_Examples

```
c
sd = socket(AF_INET, SOCK_DGRAM, 0); serv.sin_family = AF_INET;serv.sin_addr.s_addr = htonl(I
    ↪ serv));while (1) {

                            memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (inet_ntoa(cli.s
                    }
```

```
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress, port);soc
```

Observed_Examples

Related_Attack_Patterns

CAPEC-145
CAPEC-463
CAPEC-75

References

REF-18

Taxonomy_Mappings

ISA/IEC 62443: None
CLASP: None

Notes

CVEs

Template Information

ID: CVE-2023-48795

Name: OpenSSH Terrapin Attack - Detection

Severity: medium

Description: The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

Classification:

- CVSS Score: 5.9
- CVSS Metrics: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N
- CWE-ID: CWE-354
- EPSS Score: 0.69474
- EPSS Percentile: 0.97955

Vulnerability 3 - beta-updates.syneto.eu:22

CWE Information

ID

354

Name

Improper Validation of Integrity Check Value

Abstraction

Base

Structure

Simple

Status

Draft

Description

The product does not validate or incorrectly validates the integrity check values or "checksums" of a message. This may prevent it from detecting if the data has been modified or corrupted in transmission.

Extended_Description

Improper validation of checksums before use results in an unnecessary risk that can easily be mitigated. The protocol specification describes the algorithm used

for calculating the checksum. It is then a simple matter of implementing the calculation and verifying that the calculated checksum and the received checksum match. Improper verification of the calculated checksum and the received checksum can lead to far greater consequences.

Related_Weaknesses

ChildOf:345
ChildOf:345
ChildOf:754
PeerOf:353

Weakness_Ordinalities

Applicable_Platforms

Language:
Technology:

Background_Details

Alternate_Terms

Modes_Of_Introduction

Architecture and Design: None
Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

Likelihood_Of_Exploit

Medium

Common_Consequences

Integrity: Modify Application Data
Integrity: Other
Non-Repudiation: Hide Activities

Detection_Methods

Potential_Mitigations

Phase: Implementation Description: Ensure that the checksums present in messages are properly checked in accordance with the protocol specification before they are parsed and used.

Demonstrative_Examples

```
c
sd = socket(AF_INET, SOCK_DGRAM, 0); serv.sin_family = AF_INET;serv.sin_addr.s_addr = htonl(I
    ↪ serv));while (1) {

                            memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (inet_ntoa(cli.s
                    }
```

```
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress, port);soc
```

Observed_Examples

Related_Attack_Patterns

CAPEC-145
CAPEC-463
CAPEC-75

References

REF-18

Taxonomy_Mappings

ISA/IEC 62443: None
CLASP: None

Notes

CVEs

Template Information

ID: CVE-2023-48795

Name: OpenSSH Terrapin Attack - Detection

Severity: medium

Description: The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

Classification:

- CVSS Score: 5.9
- CVSS Metrics: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N
- CWE-ID: CWE-354
- EPSS Score: 0.69474
- EPSS Percentile: 0.97955

Vulnerability 4 - gw.syneto.eu:22

CWE Information

ID

354

Name

Improper Validation of Integrity Check Value

Abstraction

Base

Structure

Simple

Status

Draft

Description

The product does not validate or incorrectly validates the integrity check values or "checksums" of a message. This may prevent it from detecting if the data has been modified or corrupted in transmission.

Extended_Description

Improper validation of checksums before use results in an unnecessary risk that can easily be mitigated. The protocol specification describes the algorithm used for calculating the checksum. It is then a simple matter of implementing the calculation and verifying that the calculated checksum and the received checksum match. Improper verification of the calculated checksum and the received checksum can lead to far greater consequences.

Related_Weaknesses

ChildOf:345
ChildOf:345
ChildOf:754
PeerOf:353

Weakness_Ordinalities

Applicable_Platforms

Language:
Technology:

Background_Details

Alternate_Terms

Modes_Of_Introduction

Architecture and Design: None
Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

Likelihood_Of_Exploit

Medium

Common_Consequences

Integrity: Modify Application Data
Integrity: Other
Non-Repudiation: Hide Activities

Detection_Methods

Potential_Mitigations

Phase: Implementation Description: Ensure that the checksums present in messages are properly checked in accordance with the protocol specification before they are parsed and used.

Demonstrative_Examples

```
c
sd = socket(AF_INET, SOCK_DGRAM, 0); serv.sin_family = AF_INET;serv.sin_addr.s_addr = htonl(I
    ↪ serv));while (1) {

                            memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (inet_ntoa(cli.s
                    }
```

```
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress, port);soc
```

Observed_Examples

Related_Attack_Patterns

CAPEC-145
CAPEC-463
CAPEC-75

References

REF-18

Taxonomy_Mappings

ISA/IEC 62443: None
CLASP: None

Notes

CVEs

Template Information

ID: CVE-2023-48795

Name: OpenSSH Terrapin Attack - Detection

Severity: medium

Description: The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

Classification:

- CVSS Score: 5.9
- CVSS Metrics: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N
- CWE-ID: CWE-354
- EPSS Score: 0.69474
- EPSS Percentile: 0.97955

Vulnerability 5 - email.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 6 - lp.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 7 - email.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 8 - kb.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 9 - blog.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an en-

cryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

Vulnerability 10 - blog.syneto.eu

Template Information

ID: weak-cipher-suites

Name: Weak Cipher Suites Detection

Severity: low

Description: A weak cipher is defined as an encryption/decryption algorithm that uses a key of insufficient length. Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.

Classification:

- CVSS Score: N/A
- CVSS Metrics: N/A
- CWE-ID: N/A
- EPSS Score: N/A
- EPSS Percentile: N/A

RECON

ORIZON

Indirizzo: Crystal Palace, Via Cefalonia 70, 25124 Brescia (BS), Italia.

P.iva: IT04484750981 Email: info@orizon.one Tel: (+39) 030 0946 499