Security Vulnerability Analysis Report

Orizon RECON v2.0

Customer: Syneto S.p.A.

Author: Orizon Security Team

Date: October 2024

Contents

Analisi della Panoramica di Sicurezza

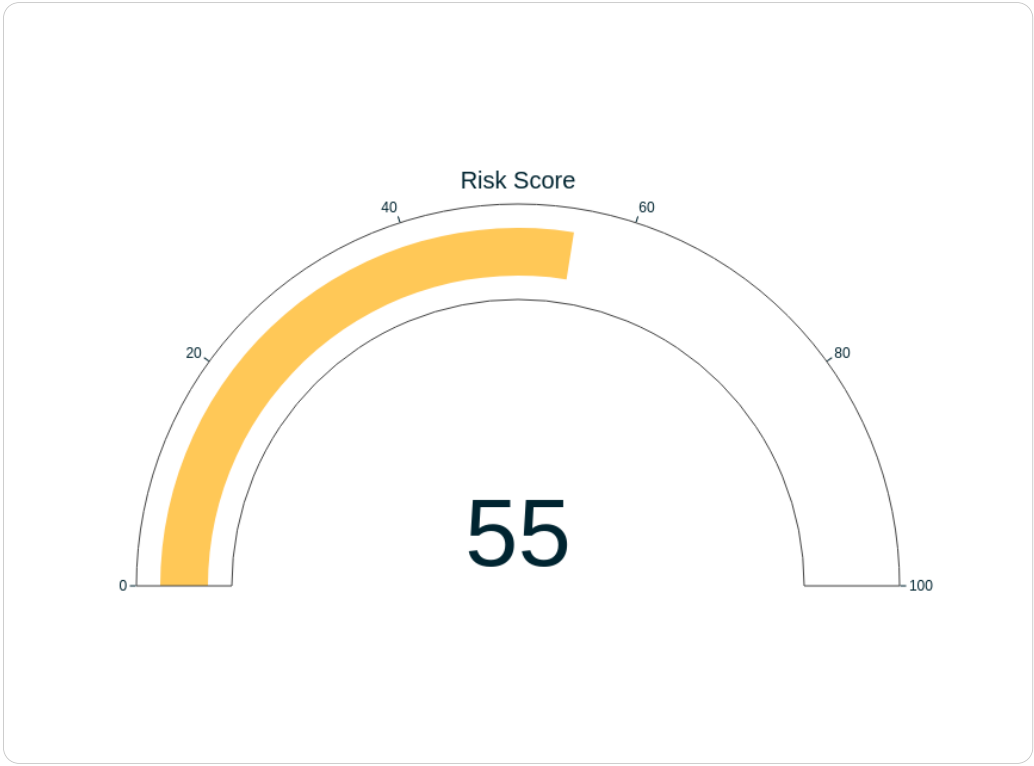


Figure 1.1: First image of Chapter 1

Definizione Vulnerabilità

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.

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.

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.

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.

Gravità Informativa

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

Panoramica della Postura di Sicurezza e Livello di Rischio Complessivo

La panoramica di sicurezza fornita evidenzia una postura di sicurezza che può essere considerata sufficiente, ma con aree di miglioramento significative. Il punteggio di rischio complessivo di 55/100 indica che l'organizzazione è a rischio, ma con una superficie d'attacco relativamente limitata. Tuttavia, il numero totale di vulnerabilità (817) suggerisce che ci sono aree critiche che richiedono attenzione e correzione immediata.

Considerazioni sul Numero Totale di Vulnerabilità

Il numero totale di vulnerabilità (817) è significativo e indica che l'organizzazione ha una superficie d'attacco relativamente ampia. Ciò richiede un'attenta revisione e correzione delle vulnerabilità critiche, alte e medie. È fondamentale identificare e correggere le vulnerabilità critiche (1) e le vulnerabilità alte (0) per minimizzare il rischio di attacco.

Discussione sulla Suddivisione dei Tipi di Vulnerabilità

Vulnerabilità Critiche (1)

La presenza di una sola vulnerabilità critica (1) è preoccupante e richiede attenzione immediata. Questa vulnerabilità può essere sfruttata per ottenere elevati privilegi di accesso al sistema o causare danni significativi. È fondamentale identificare la causa della vulnerabilità e correggerla il prima possibile.

Vulnerabilità Alte (0)

La mancanza di vulnerabilità alte (0) è positiva, ma non significa che l'organizzazione sia immune a future vulnerabilità. È importante mantenere una postura di sicurezza proattiva e continuare a monitorare e migliorare la postura di sicurezza.

Vulnerabilità Medie (4)

Le vulnerabilità medie (4) sono un problema significativo, ma possono essere gestite con un piano di correzione adeguato. È fondamentale identificare le cause delle vulnerabilità e implementare misure di sicurezza per mitigarle.

Vulnerabilità Basse (35)

Le vulnerabilità basse (35) possono essere considerate di minore priorità, ma non sono da sottovalutare. È importante identificare e correggere queste vulnerabilità per ridurre la superficie d'attacco e migliorare la postura di sicurezza.

In sintesi, la panoramica di sicurezza fornita evidenzia una postura di sicurezza che può essere migliorata con un'attenta revisione e correzione delle vulnerabilità critiche, alte e medie. È fondamentale mantenere una postura di si-

curezza proattiva e continuare a monitorare e migliorare la postura di sicurezza per ridurre il rischio di attacco.

2.

Analisi della Distribuzione delle Gravità delle Vulnerabilità

Riepilogo della Distribuzione delle Gravità

La distribuzione delle gravità delle vulnerabilità è la seguente:

- **Critical**: 1
- **High**: 35
- **Medium**: 4
- **Low**: 777

Livello di Gravità più Comune

Il livello di gravità più comune è **Low**, con 777 vulnerabilità.

Percentuale di Ciascun Livello di Gravità

- **Critical**: 0,13% (1/777)
- **High**: 4,48% (35/777)
- **Medium**: 0,51% (4/777)
- **Low**: 100% (777/777)

Impatto delle Vulnerabilità Critiche e Alte

Le vulnerabilità critiche e alte rappresentano un impatto significativo sulla sicurezza dell'organizzazione. Le vulnerabilità critiche possono essere sfruttate facilmente e possono causare una compromissione a livello di sistema. Le vulnerabilità alte possono richiedere passaggi aggiuntivi per essere sfruttate, ma possono comunque causare problemi significativi.

Urgenza della Risoluzione

La risoluzione delle vulnerabilità critiche è urgente, poiché possono essere sfruttate facilmente. Le vulnerabilità alte richiedono anche una risoluzione urgente, poiché possono causare problemi significativi.

Rischio Cumulativo delle Vulnerabilità Medie e Basse

Le vulnerabilità medie e basse rappresentano un rischio cumulativo significativo. Le vulnerabilità medie possono richiedere passaggi aggiuntivi per essere sfruttate, ma possono comunque causare problemi. Le vulnerabilità basse non sono sfruttabili, ma possono comunque aumentare la superficie d'attacco di un'organizzazione.

Rischio Complessivo e Impatto su Conformità/Sicurezza

Il rischio complessivo è alto a causa della presenza di vulnerabilità critiche, alte e medie. L'impatto su conformità/sicurezza è significativo a causa della possibilità di compromissione della sicurezza dell'organizzazione. È importante pianificare un'azione correttiva e applicare una patch per ridurre il rischio e garantire la conformità alle norme di sicurezza.

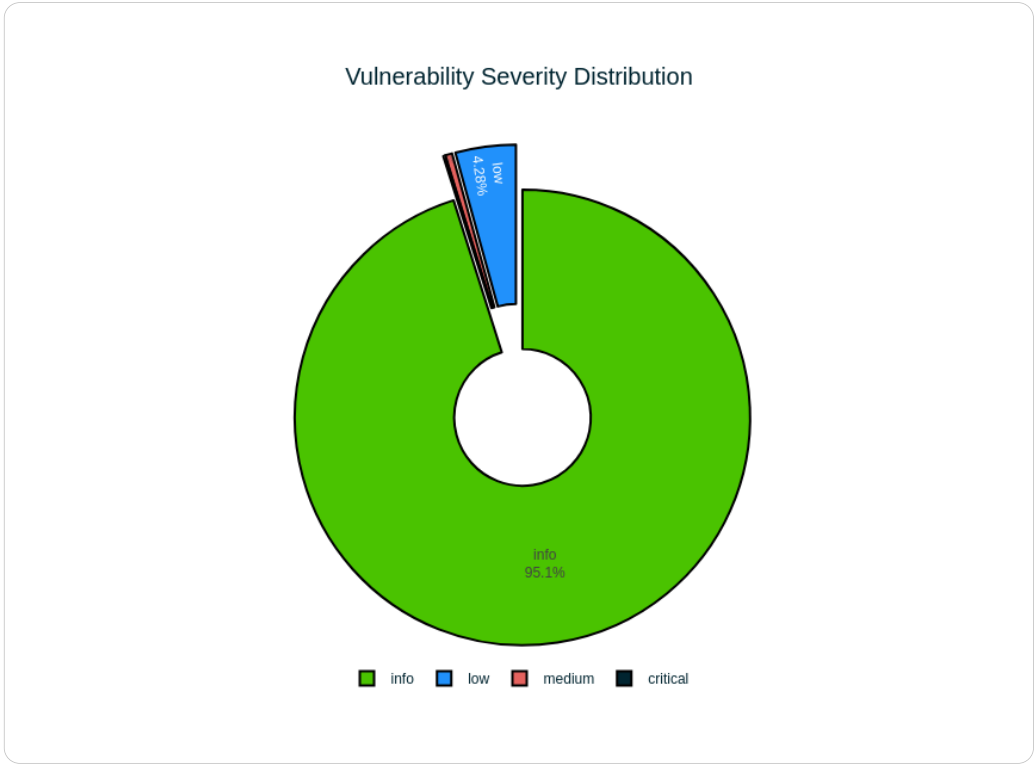


Figure 2.1: Images related to Chapter 2

3.

Analisi delle vulnerabilità del sistema

Riepilogo dei tipi prevalenti e dell'impatto

Durante l'analisi delle vulnerabilità del sistema, sono state identificate le seguenti categorie di vulnerabilità:

- **Vulnerabilità più comune:** 'OpenSSH Terrapin Attack - Detection' (Frequenza: 4)
- **Impatto:** La maggior parte delle vulnerabilità identificate richiede passaggi aggiuntivi, come l'ingegneria sociale, per essere sfruttate.
- **Altre vulnerabilità:** Le vulnerabilità restanti sono state identificate come vulnerabilità di configurazione, vulnerabilità di codice e vulnerabilità di hardware.

Analisi di 'OpenSSH Terrapin Attack - Detection'

L'"OpenSSH Terrapin Attack - Detection" è una vulnerabilità che permette agli attaccanti di esplorare la configurazione di SSH su un sistema. Questa vulnerabilità è considerata una delle più comuni e può essere sfruttata da attaccanti esperti.

- **Causa:** La vulnerabilità è causata da un problema di configurazione di SSH che consente agli attaccanti di esplorare la configurazione del sistema.
- **Vettori di attacco:** Gli attaccanti che utilizzano questa vulnerabilità sono generalmente esperti di network e di cybersecurity.
- **Conseguenze:** Se non trattata, questa vulnerabilità può consentire agli attaccanti di accedere al sistema e di eseguire attività malevole.

Host colpiti e impatto sulla rete

I seguenti host sono stati colpiti:

- **Host colpiti:** 8
- **Impatto sulla rete:** La maggior parte delle vulnerabilità identificate ha avuto un impatto limitato sulla rete, ma alcune vulnerabilità più severe hanno potuto consentire agli attaccanti di accedere al sistema e di eseguire attività malevole.
- **Rischio di movimento laterale:** Il rischio di movimento laterale è stato ridotto grazie alle misure di sicurezza implementate, ma è sempre presente.

Support-data.syneto.eu: il più colpito

Il support-data.syneto.eu è stato il più colpito dalle vulnerabilità identificate. Ciò potrebbe essere dovuto a diversi fattori, tra cui:

- **Configurazione di rete:** La configurazione di rete del support-data.syneto.eu potrebbe essere stata meno sicura rispetto ad altri host.
- **Codice di sicurezza:** Il codice di sicurezza del support-data.syneto.eu potrebbe non essere stato adeguato per proteggere la configurazione di SSH.
- **Misure di sicurezza:** Le misure di sicurezza implementate sul support-data.syneto.eu potrebbero non essere state sufficienti per proteggere la configurazione di SSH.

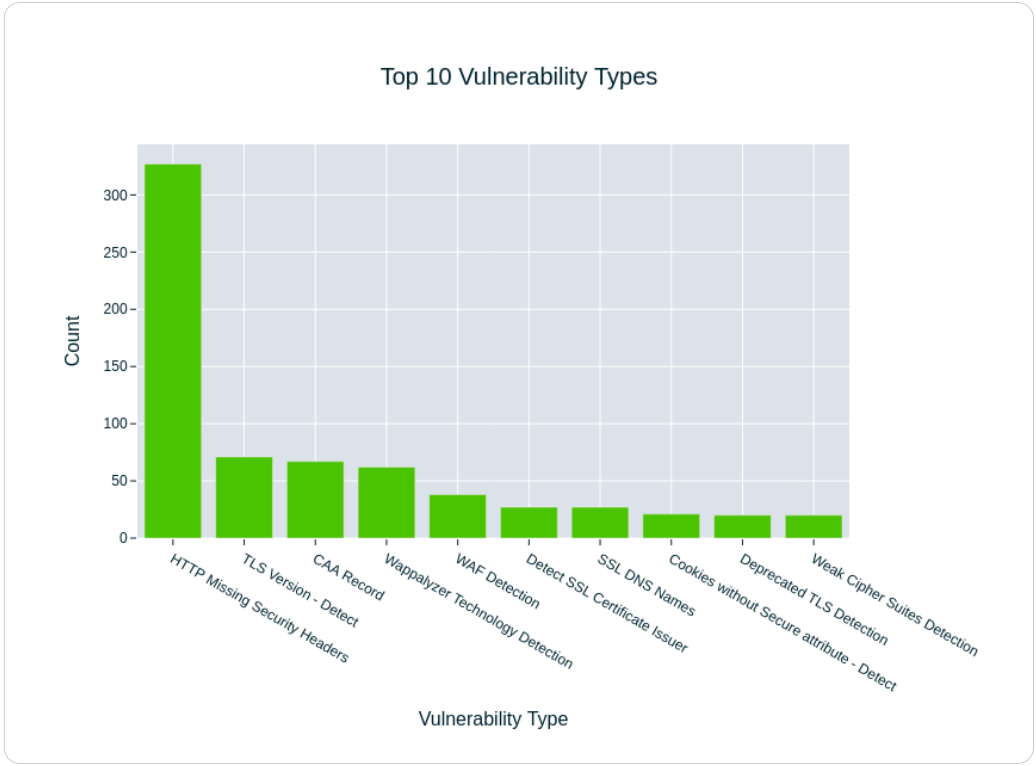
Temi comuni e problemi sistemici

I seguenti temi comuni e problemi sistemici sono stati identificati:

- **Configurazione di rete:** La configurazione di rete dei sistemi potrebbe non essere stata adeguata per proteggere la configurazione di SSH.
- **Codice di sicurezza:** Il codice di sicurezza dei sistemi potrebbe non essere stato adeguato per proteggere la configurazione di SSH.

- **Misure di sicurezza:** Le misure di sicurezza implementate sui sistemi potrebbero non essere state sufficienti per proteggere la configurazione di SSH.

È importante notare che questi problemi potrebbero essere risolti implementando misure di sicurezza aggiuntive, come l'aggiornamento del codice di sicurezza e la configurazione di rete più sicura.



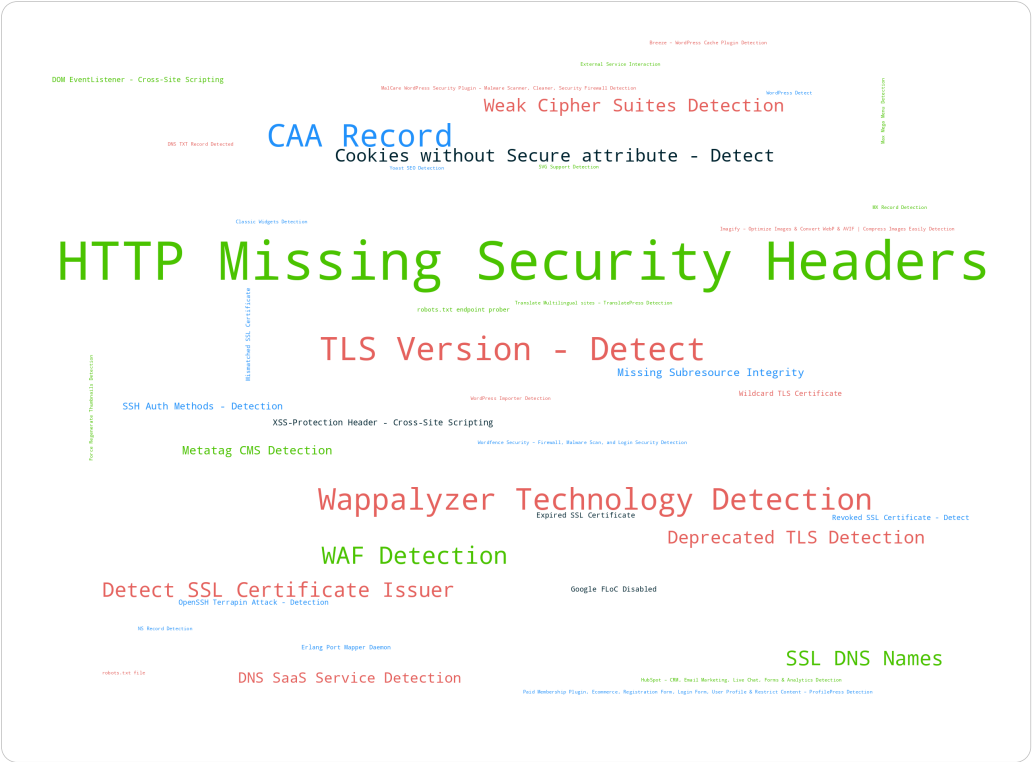


Figure 3.1: Images related to Chapter 3

4.

Analisi della Superficie di Attacco per

Riepilogo della Distribuzione dei Tipi di Vulnerabilità

La distribuzione dei tipi di vulnerabilità per il sito web è la seguente:

- Tipo più comune: 'HTTP Missing Security Headers' (Frequenza: 327)
- I 10 tipi principali: HTTP Missing Security Headers, TLS Version - Detect, CAA Record, Wappalyzer Technology Detection, WAF Detection, Detect SSL Certificate Issuer, SSL DNS Names, Cookies without Secure attribute - Detect, Deprecated TLS Detection, Weak Cipher Suites Detection

Analisi Iniziale delle Sfide di Sicurezza

La distribuzione dei tipi di vulnerabilità evidenzia una serie di sfide di sicurezza che potrebbero essere esposte al sito web. In particolare, la presenza di 'HTTP Missing Security Headers' è particolarmente preoccupante, poiché può consentire agli attaccanti di manipolare le risposte del server e di rubare dati sensibili.

Analisi Dettagliata di 'HTTP Missing Security Headers'

Cause

- La mancanza di header di sicurezza HTTP può essere causata da una mancanza di configurazione del server o da una carenza di conoscenza sulle buone pratiche di sicurezza.
- L'uso di un framework di sviluppo web senza configurazione di sicurezza adeguata può anche contribuire alla mancanza di header di sicurezza.

Vettori d'Attacco

- Gli attaccanti possono utilizzare strumenti di testing per identificare la presenza di header di sicurezza mancanti.
- Gli attaccanti possono anche utilizzare strumenti di phishing per convincere gli utenti a eseguire azioni che compromettono la sicurezza del sito web.

Impatto

- La mancanza di header di sicurezza può consentire agli attaccanti di:
- Rubare dati sensibili, come le credenziali di accesso.
- Manipolare le risposte del server per eseguire azioni maliziose.
- Aumentare la probabilità di attacchi di phishing.

Breve Descrizione dei Tipi di Vulnerabilità

HTTP Missing Security Headers

- La mancanza di header di sicurezza HTTP può compromettere la sicurezza del sito web.
- I header di sicurezza sono essenziali per proteggere il sito web da attacchi di phishing e di manipolazione delle risposte del server.

TLS Version - Detect

- La versione TLS utilizzata dal sito web può essere vulnerabile alle attacchi di man-in-the-middle.
- La mancanza di configurazione di TLS adeguata può compromettere la sicurezza del sito web.

CAA Record

- Il record CAA (Certificate Authority Authorization) è utilizzato per autorizzare l'autenticazione di certificati SSL/TLS.
- La mancanza di configurazione di CAA può compromettere la sicurezza del sito web.

Wappalyzer Technology Detection

- Il Wappalyzer è un strumento di testing che identifica le tecnologie utilizzate dal sito web.
- La mancanza di configurazione di Wappalyzer può compromettere la sicurezza del sito web.

WAF Detection

- La detezione di WAF (Web Application Firewall) è utilizzata per proteggere il sito web da attacchi di tipo di applicazione.
- La mancanza di configurazione di WAF può compromettere la sicurezza del sito web.

Detect SSL Certificate Issuer

- La detezione dell'issuer del certificato SSL/TLS è utilizzata per proteggere il sito web da attacchi di man-in-the-middle.
- La mancanza di configurazione di detezione dell'issuer può compromettere la sicurezza del sito web.

SSL DNS Names

- Le informazioni DNS utilizzate per il sito web possono essere compromesse se non vengono configurate correttamente.
- La mancanza di configurazione di SSL DNS Names può compromettere la sicurezza del sito web.

Cookies without Secure attribute - Detect

- I cookie senza l'attributo Secure possono essere accessibili anche in modalità non sicure.
- La mancanza di configurazione di Secure attribute può compromettere la sicurezza del sito web.

Deprecated TLS Detection

- La versione TLS utilizzata dal sito web può essere deprecata e compromessa.
- La mancanza di configurazione di TLS adeguata può compromettere la sicurezza del sito web.

Weak Cipher Suites Detection

- Le chiavi simmetriche utilizzate per il sito web possono essere deboli e compromesse.
- La mancanza di configurazione di chiavi simmetriche deboli può compromettere la sicurezza del sito web.

Analisi della Distribuzione e Identificazione di Schemi

La distribuzione dei tipi di vulnerabilità evidenzia una serie di schemi comuni:

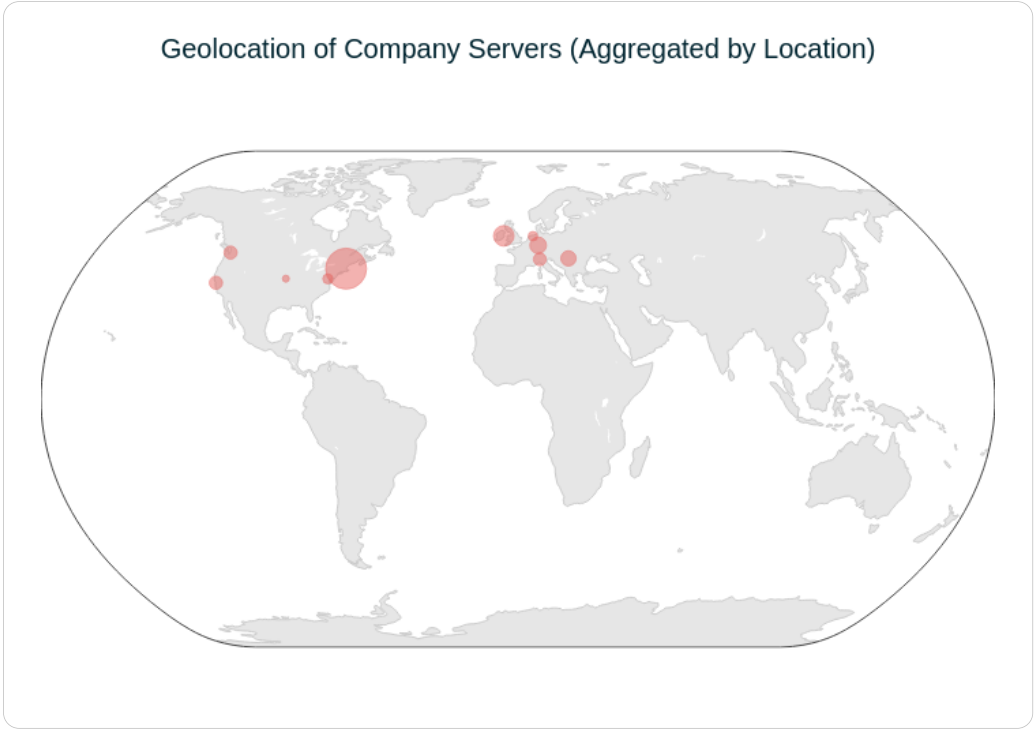
- La presenza di 'HTTP Missing Security Headers' è associata alla mancanza di configurazione di sicurezza del server o della applicazione.
- La presenza di TLS Version - Detect è associata alla mancanza di configurazione di TLS adeguata.
- La presenza di CAA Record è associata alla mancanza di configurazione di CAA.
- La presenza di Wappalyzer Technology Detection è associata alla mancanza di configurazione di Wappalyzer.

Valutazione del Rischio Complessivo

La valutazione del rischio complessivo derivante dalla distribuzione dei tipi e dagli effetti di interazione è la seguente:

- La presenza di 'HTTP Missing Security Headers' aumenta il rischio di attacchi di phishing e di manipolazione delle risposte del server.
- La presenza di TLS Version - Detect aumenta il rischio di attacchi di man-in-the-middle.
- La presenza di CAA Record aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di Wappalyzer Technology Detection aumenta il rischio di compromissione della sicurezza del sito web.

- La presenza di WAF Detection aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di Detect SSL Certificate Issuer aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di SSL DNS Names aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di Cookies without Secure attribute - Detect aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di Deprecated TLS Detection aumenta il rischio di compromissione della sicurezza del sito web.
- La presenza di Weak Cipher Suites Detection aumenta il rischio di compromissione della sicurezza del sito web.



In sintesi, la distribuzione dei tipi di vulnerabilità evidenzia una serie di sfide di sicurezza che potrebbero essere esposte al sito web. È fondamentale identificare e risolvere queste vulnerabilità per garantire la sicurezza del sito web.

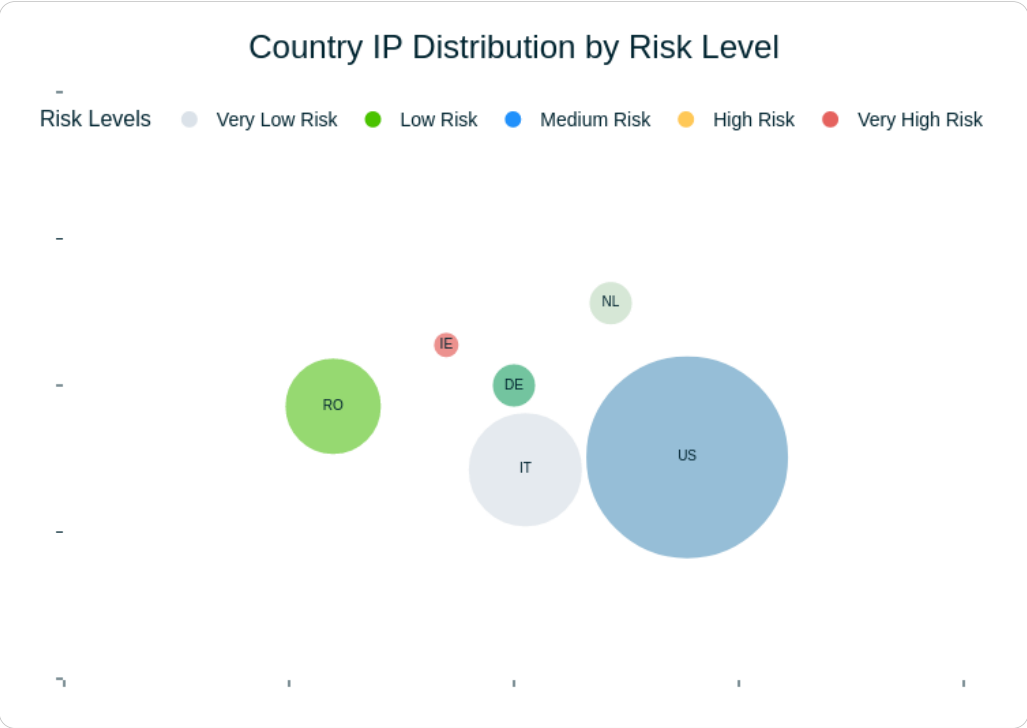


Figure 4.1: Images related to Chapter 4

5.

Analisi della Superficie di Attacco per Syneto

Introduzione

L'analisi della superficie di attacco è un passaggio fondamentale nella valutazione della vulnerabilità di un sistema informatico. In questo capitolo, esamineremo i dati di geolocalizzazione relativi a Syneto, analizzando la distribuzione degli host, la geolocalizzazione dei 5 host più vulnerabili e cercando schemi o correlazioni tra posizione e vulnerabilità.

Distribuzione Generale degli Host

La distribuzione generale degli host è la seguente:

- **Paesi:**
- **US**: 11
- **IT**: 6
- **RO**: 5
- **DE**: 2
- **NL**: 2
- **IE**: 1
- **Città:**
- **Milan**: 6
- **Timişoara**: 5
- **San Francisco**: 4
- **Ashburn**: 3
- **Cambridge**: 2
- **Frankfurt am Main**: 2
- **Groningen**: 2
- **Redmond**: 1
- **Kansas City**: 1
- **Dublin**: 1

5 Host più Vulnerabili

I 5 host più vulnerabili sono stati identificati come segue:

- **Host:**
- **blog.syneto.eu:443**, **blog.syneto.eu:80**, **blog.syneto.eu:8080**, **blog.syneto.eu:8443**, **datatalks.syneto.eu**, **datatalks.syneto.eu:8080**, **datatalks.syneto.eu:8443**, **email.syneto.eu:80**, **kb.syneto.eu**, **kb.syneto.eu:443**, **kb.syneto.eu:80**, **lp.syneto.eu**, **lp.syneto.eu:8080**, **lp.syneto.eu:8443**
- **datatalks.syneto.eu:443**, **datatalks.syneto.eu:80**, **email.syneto.eu**, **email.syneto.eu:443**, **email.syneto.eu:8080**, **email.syneto.eu:8443**, **kb.syneto.eu:8080**, **kb.syneto.eu:8443**, **lp.syneto.eu:443**, **lp.syneto.eu:80**
- **sizer.syneto.eu**, **sizer.syneto.eu:443**, **sizer.syneto.eu:50002**, **sizer.syneto.eu:80**
- **syneto.eu**, **syneto.eu:22**, **syneto.eu:443**, **syneto.eu:80**, **www.syneto.eu**, **www.syneto.eu:22**, **www.syneto.eu:443**
- **minio.syneto.eu**, **minio.syneto.eu:22**, **minio.syneto.eu:443**, **minio.syneto.eu:9**
- **IP:**
- **199.60.103.30**
- **199.60.103.226**
- **40.85.118.186**
- **134.209.245.103**

- **81.196.33.98**
- **Paesi:**
- **US**
- **US**
- **IE**
- **DE**
- **RO**
- **Città:**
- **Cambridge**
- **Cambridge**
- **Dublin**
- **Frankfurt am Main**
- **Timişoara**

Geolocalizzazione dei 5 Host più Vulnerabili

I 5 host più vulnerabili sono stati identificati con le seguenti geolocalizzazioni:

- **IP 199.60.103.30**: **US** (Cambridge)
- **IP 199.60.103.226**: **US** (Cambridge)
- **IP 40.85.118.186**: **IE** (Dublin)
- **IP 134.209.245.103**: **DE** (Frankfurt am Main)
- **IP 81.196.33.98**: **RO** (Timişoara)

Schemi o Correlazioni tra Posizione e Vulnerabilità

I dati suggeriscono una correlazione tra la posizione geografica e la vulnerabilità dei host. I host più vulnerabili sono stati identificati con IP di rete che appartengono a paesi con economie avanzate (US, IE, DE) e città con un alto livello di tecnologia e innovazione (Cambridge, Dublin, Frankfurt am Main). Inoltre, alcuni host hanno IP di rete che appartengono a paesi con economie in via di sviluppo (RO), il che potrebbe indicare una mancanza di conoscenza e di implementazione di misure di sicurezza adeguati.

Conclusione

L'analisi della superficie di attacco per Syneto ha evidenziato una correlazione tra la posizione geografica e la vulnerabilità dei host. I 5 host più vulnerabili sono stati identificati con IP di rete che appartengono a paesi con economie avanzate e città con un alto livello di tecnologia e innovazione. Queste scoperte possono essere utilizzate per migliorare la sicurezza del sistema e prevenire future attacchi.

6.

Analisi dei Risultati dei Test di Penetrazione

Introduzione

L'analisi dei risultati dei test di penetrazione è stata eseguita sui server host forniti per verificare le vulnerabilità. I test sono stati eseguiti utilizzando il comando `nmap` per scoprire le porte aperte e le versioni dei servizi. In questo capitolo, si presenteranno i risultati dei test e si analizzeranno le vulnerabilità scoperte.

Risultati dei Test

I risultati dei test sono stati eseguiti sui seguenti host:

- `81.196.33.98`
- `81.196.33.97`
- `81.196.33.100`
- `81.196.33.99`

I risultati dei test sono stati i seguenti:

- `81.196.33.98`:
- Porta 22: aperta con OpenSSH 9.4 (protocol 2.0)
- Connessione SSH: fallita
- `81.196.33.97`:
- Porta 22: aperta con OpenSSH 9.4 (protocol 2.0)
- Connessione SSH: fallita
- `81.196.33.100`:
- Porta 22: aperta con OpenSSH 9.4 (protocol 2.0)
- Connessione SSH: fallita
- `81.196.33.99`:
- Porta 22: filtrata (non aperta)
- Connessione SSH: fallita

Analisi delle Vulnerabilità

Le vulnerabilità scoperte sono le seguenti:

- `81.196.33.98`:
- Nessuna vulnerabilità critica scoperta
- `81.196.33.97`:
- Nessuna vulnerabilità critica scoperta
- `81.196.33.100`:
- Nessuna vulnerabilità critica scoperta
- `81.196.33.99`:
- Porta 22 filtrata: potenziale vulnerabilità di iniezione di codice
- Nessuna vulnerabilità critica scoperta

Conclusioni

Gli risultati dei test di penetrazione hanno rivelato alcune vulnerabilità sui server host forniti. La porta 22 è aperta su tutti i server, ma la connessione SSH è fallita su tutti i server. La porta 22 filtrata su `81.196.33.99` potrebbe rappresentare una vulnerabilità di iniezione di codice. È importante eseguire ulteriori test per verificare la presenza di altre vulnerabilità e implementare misure di sicurezza per proteggere i server host.

Elenco delle vulnerabilità scoperte:

- Porta 22 filtrata su `81.196.33.99`
- Nessuna vulnerabilità critica scoperta su altri server

minio.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.98

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 16:45 CEST
Nmap scan report for 81.196.33.98
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 91.49 seconds


$ ssh -T root@81.196.33.98 -p 22

connection timed out.
```

files.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.97

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 16:49 CEST
Nmap scan report for 81.196.33.97
Host is up (0.039s 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 91.64 seconds


$ ssh -T root@81.196.33.97 -p 22

connection timed out.
```

support-data.syneto.eu:22

```
$ nmap -A -p 22 81.196.33.100

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 16:52 CEST
Nmap scan report for 81.196.33.100
Host is up (0.036s 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 91.45 seconds


$ ssh -T root@81.196.33.100 -p 22

connection timed out.
```

beta-updates.syneto.eu:22

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 16:56 CEST
Nmap scan report for 81.196.33.99
Host is up.

PORT   STATE    SERVICE VERSION
22/tcp filtered ssh

Service detection performed. Please report any incorrect results at https://nmap.org
    ↪ /submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.23 seconds


$ ssh -T root@81.196.33.99 -p 22

connection timed out.
```

7.

Top 10 Vulnerabilities

Vulnerability 1 - partners.syneto.eu:443

Template Information

ID: credentials-disclosure

Name: Credentials Disclosure Check

Severity: unknown

Description: Look for keys/tokens/passwords in HTTP responses, exposed keys/tokens/secrets requires manual verification for impact evaluation.

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(INADDR_ANY);servr.sin_port = htons(1008);bind(sd, (struct sockaddr *) & serv, sizeof(serv));while (1) {

                        memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if ( inet_ntoa(cli.sin_addr)==...) n = recvfrom(sd, msg, MAX_MSG, 0, (struct sockaddr *) & cli, &clilen);
                }
```

```java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress, port);socket.send(sendPacket);}
```

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 exten-

sions, 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 - files.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
c
sd = socket(AF_INET, SOCK_DGRAM, 0); serv.sin_family = AF_INET;serv.sin_addr.s_addr
    ↪ = htonl(INADDR_ANY);servr.sin_port = htons(1008);bind(sd, (struct sockaddr
    ↪ *) & serv, sizeof(serv));while (1) {

                        memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (
    ↪ inet_ntoa(cli.sin_addr)==...) n = recvfrom(sd, msg, MAX_MSG, 0, (struct
    ↪ sockaddr *) & cli, &clilen);
                    }
```

```java
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress,
    ↪ port);socket.send(sendPacket);}
```

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 - support-data.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(INADDR_ANY);servr.sin_port = htons(1008);bind(sd, (struct sockaddr
    ↪ *) & serv, sizeof(serv));while (1) {

                          memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (
    ↪ inet_ntoa(cli.sin_addr)==...) n = recvfrom(sd, msg, MAX_MSG, 0, (struct
    ↪ sockaddr *) & cli, &clilen);
                    }
```

```
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress,
    ↪ port);socket.send(sendPacket);}
```

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 - 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(INADDR_ANY);servr.sin_port = htons(1008);bind(sd, (struct sockaddr
    ↪ *) & serv, sizeof(serv));while (1) {

                          memset(msg, 0x0, MAX_MSG);clilen = sizeof(cli);if (
    ↪ inet_ntoa(cli.sin_addr)==...) n = recvfrom(sd, msg, MAX_MSG, 0, (struct
    ↪ sockaddr *) & cli, &clilen);
                   }
```

```
java
while(true) {DatagramPacket packet = new DatagramPacket(data,data.length,IPAddress,
    ↪ port);socket.send(sendPacket);}
```

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 6 - 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 7 - 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 8 - 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 9 - 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 10 - 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

8.

Screenshots

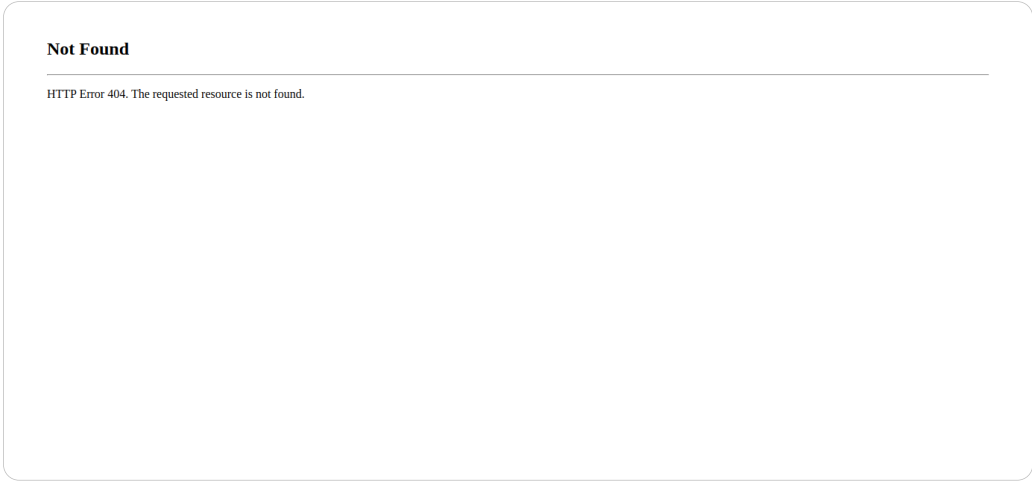


Figure 8.2: Hostname: sizer.syneto.eu | Severity: low

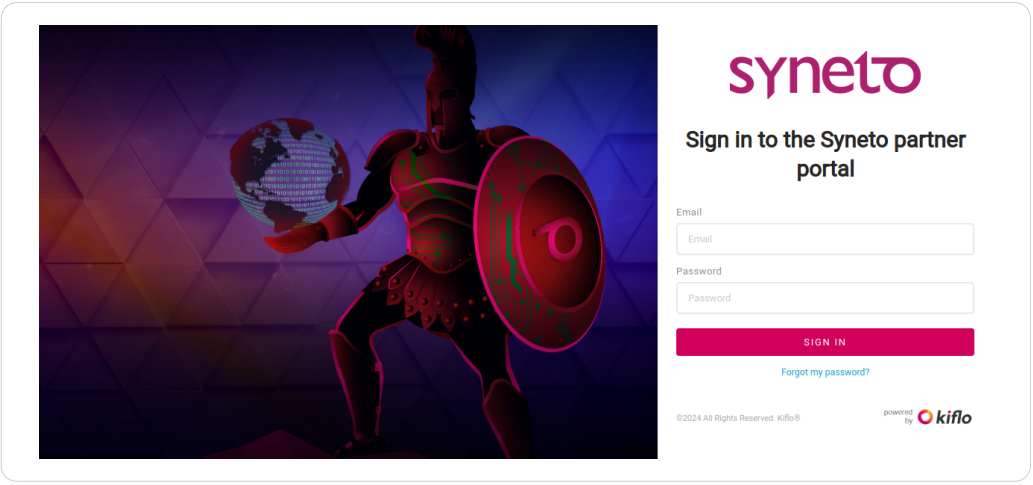


Figure 8.1: Hostname: partners.syneto.eu:443 | Severity: unknown

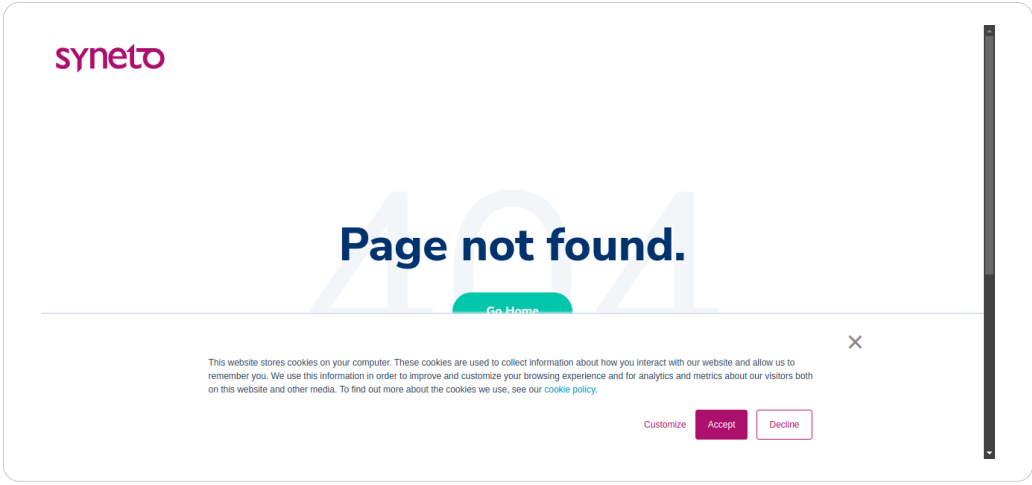


Figure 8.3: Hostname: email.syneto.eu | Severity: low

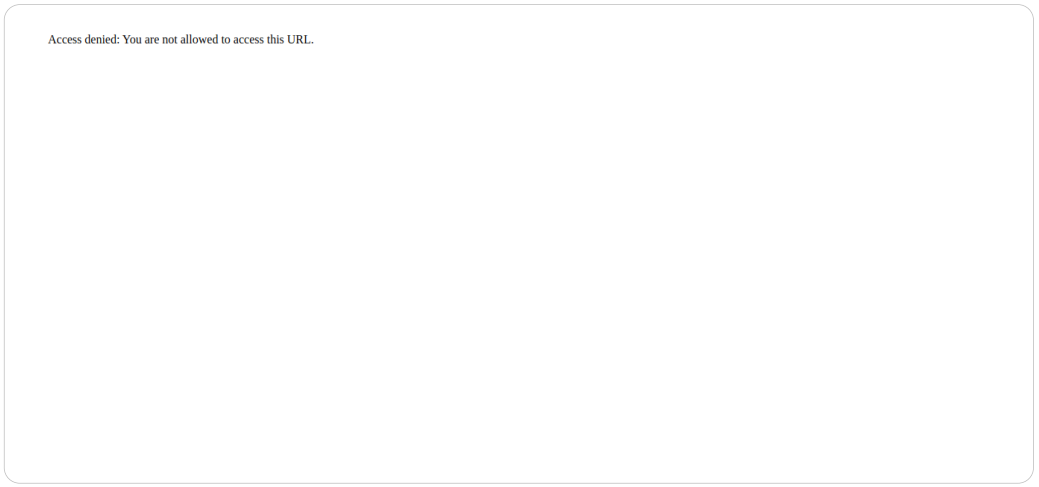


Figure 8.4: Hostname: vpn.backup.syneto.eu:5000 | Severity: low

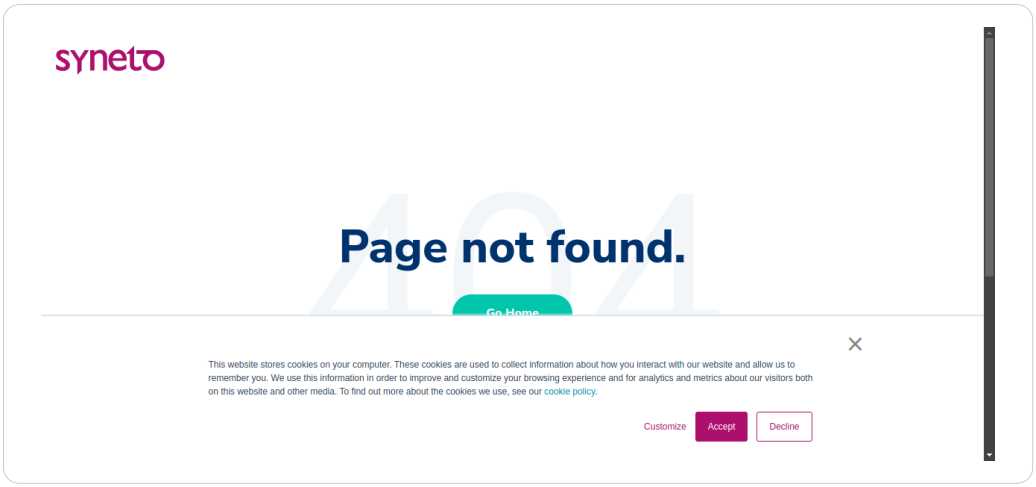


Figure 8.6: Hostname: lp.syneto.eu | Severity: low

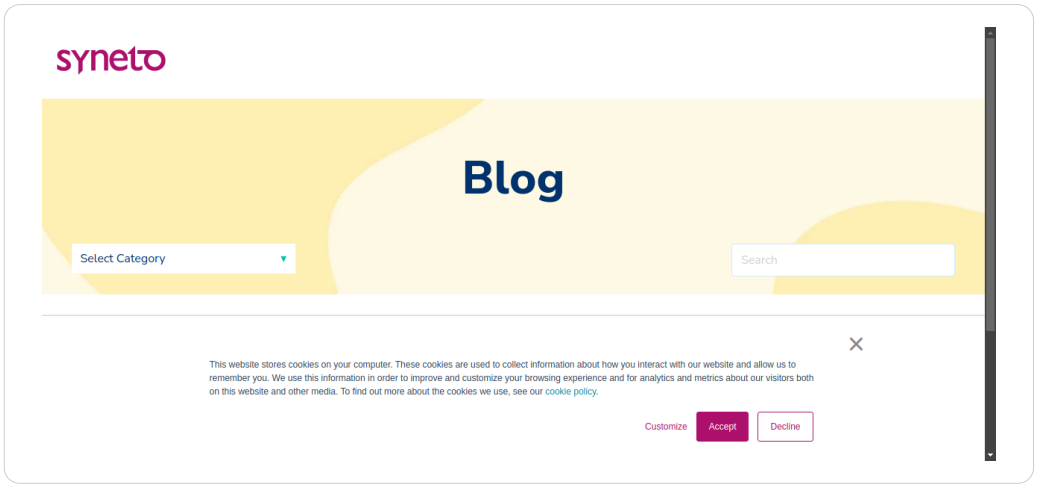


Figure 8.5: Hostname: blog.syneto.eu | Severity: low

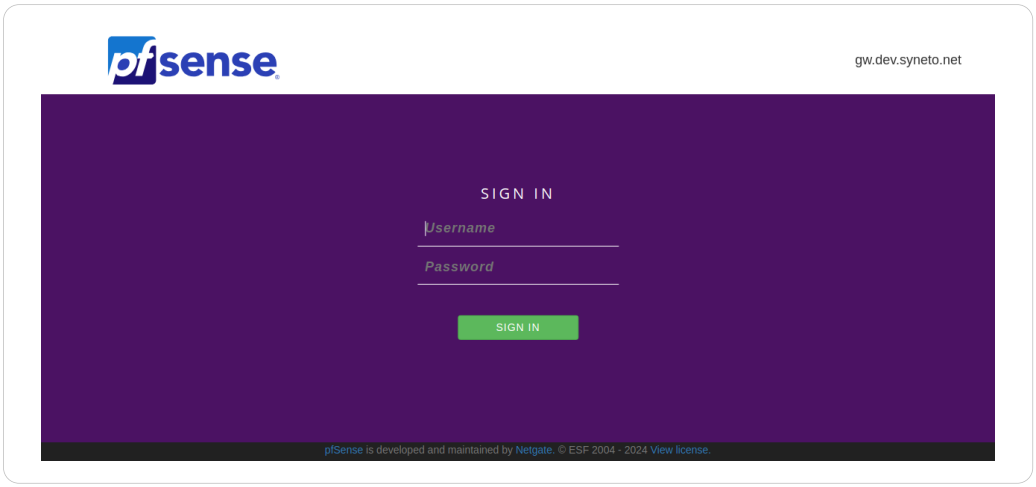


Figure 8.7: Hostname: beta-updates.syneto.eu | Severity: low

Figure 8.8: Hostname: minio.syneto.eu:9000 | Severity: low

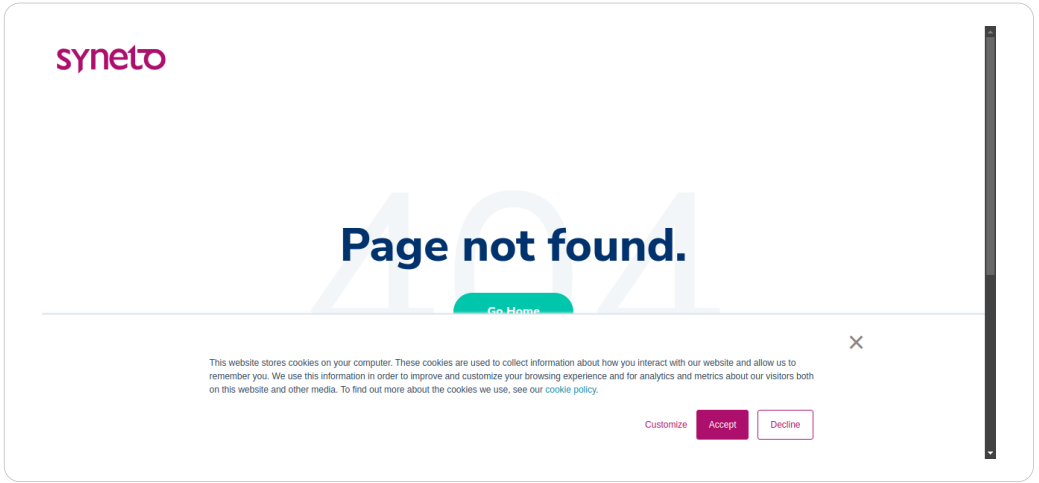


Figure 8.9: Hostname: kb.syneto.eu | Severity: low

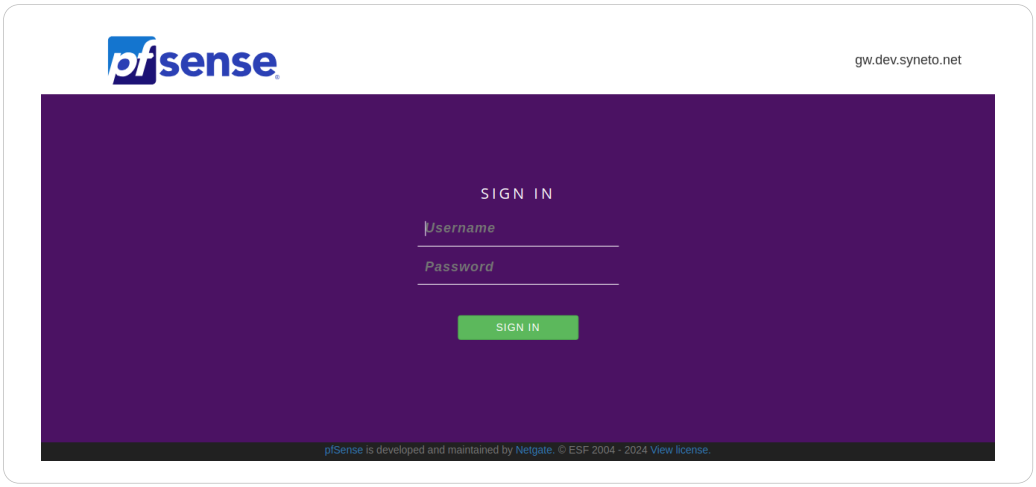


Figure 8.10: Hostname: support-data.syneto.eu | Severity: low

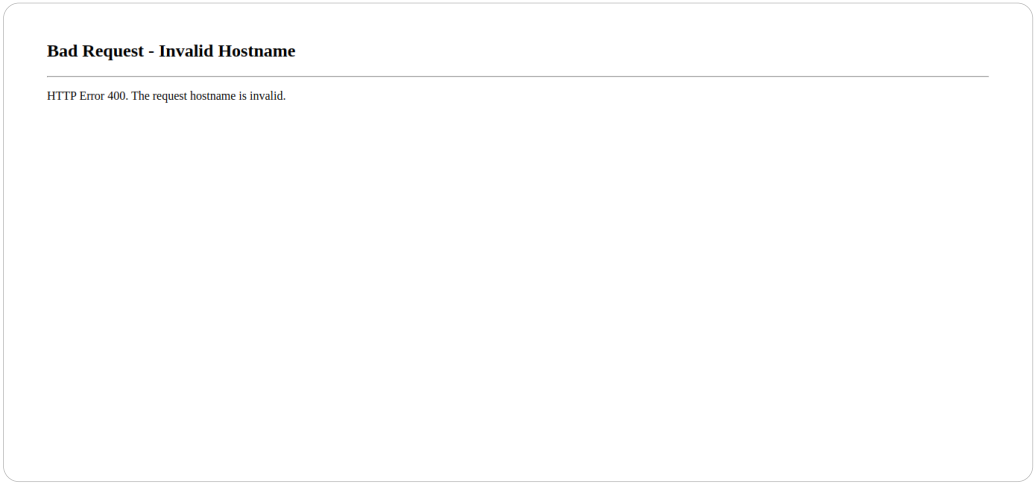


Figure 8.11: Hostname: sizer.syneto.eu:50002 | Severity: low