Muninn

Network Encryption Threat Report

www.muninn.ai

Table of contents

Table of contents

Introduction

by Henrik Falkenthros
Senior IT Security Engineer
https://www.linkedin.com/in/henrikfalkenthros

Cyber security has been a rapidly growing concern for companies, governments, and citizens for well over a decade. Nation states continue to use cyber weapons to sabotage their adversaries, and this tendency has increased drastically since Russia invaded Ukraine February 24th, 2022.

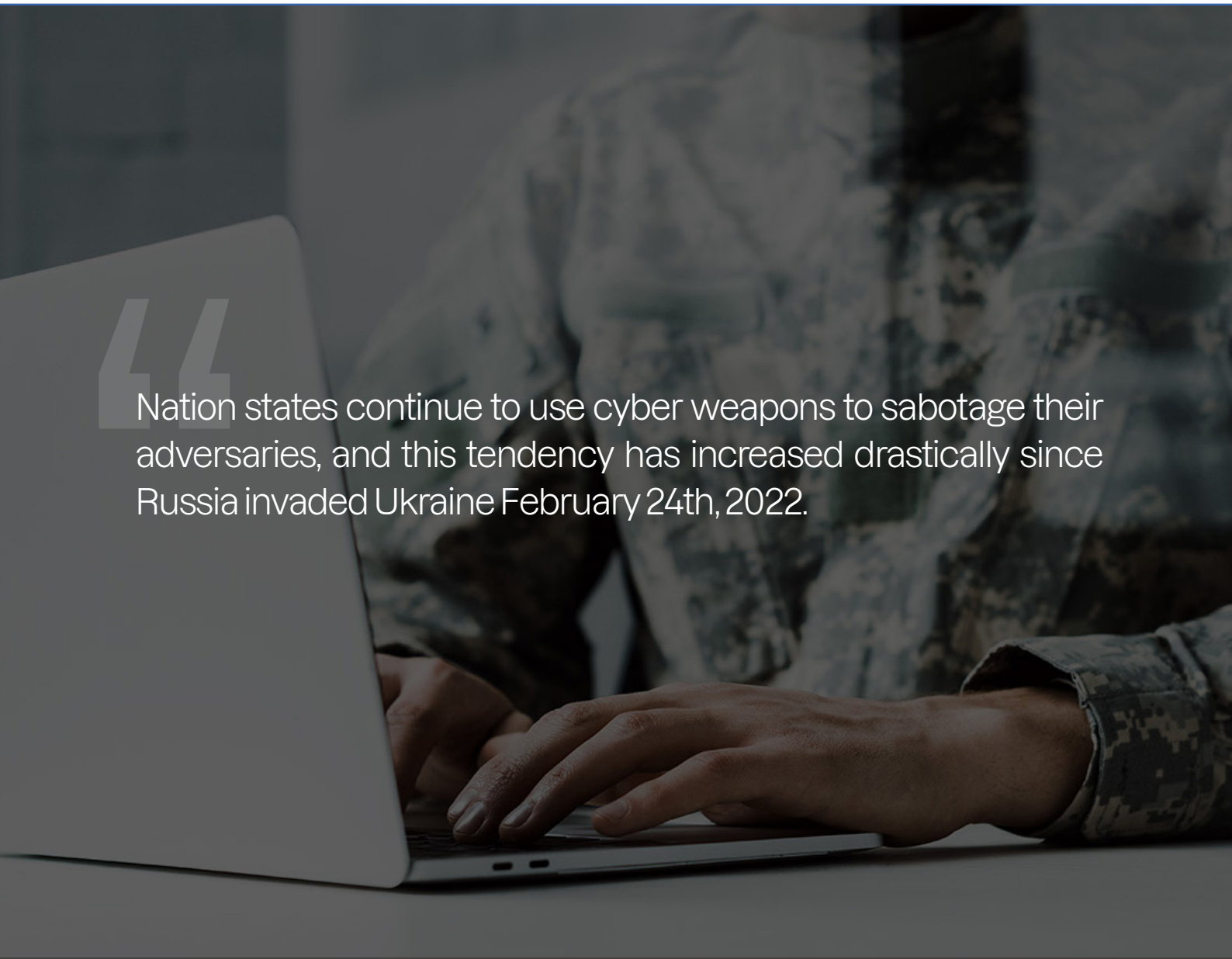
In the following Threat Report, we will take a closer look at the types of cyber-attacks Muninn sensors around the World have detected in the wake of Russia's invasion of Ukraine.

Cyber-attacks come in many different shapes and sizes, and it is difficult to identify the threat actor responsible for perpetrating a cyber-attack – especially when the attackers are state sponsored cyber-gangs. However, our customers have noticed an increase in notifications originating from Russian or Belarusian IP addresses after the Russian invasion of Ukraine on February 24, 2022.

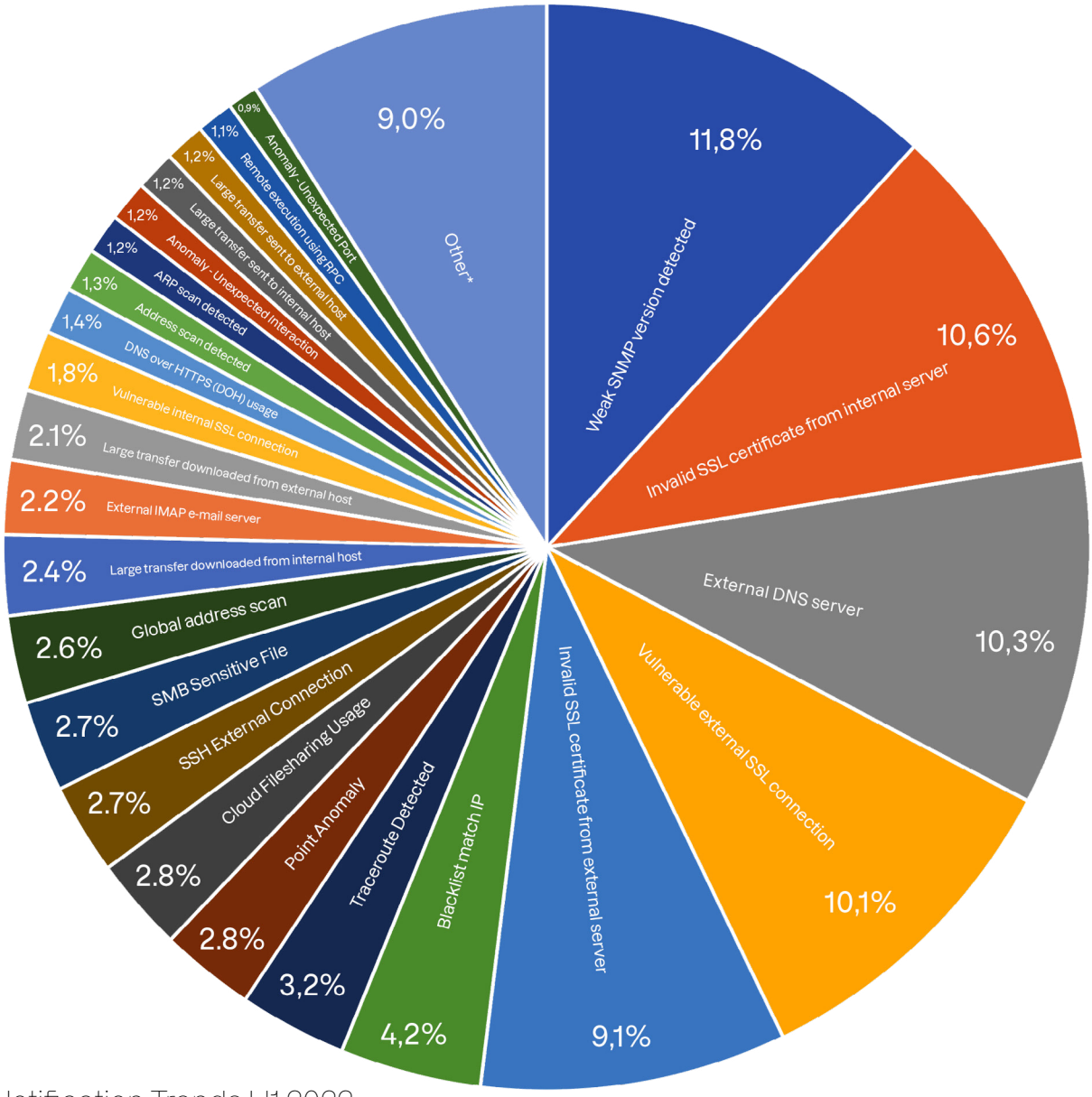
H1 2022 notification category distribution

Muninn network sensors have ingested and analysed enormous amounts of network data across the globe and generated threat notifications that have helped many companies evade costly cyber incidents.

The following section provides a statistical overview of the types and quantity of threats Muninn sensors have detected across the entire customer base.



Top 50 Notifications in H1 2022



Notification Trends H1 2022

Muninn sensors detected a notedly increase in "Weak SNMP versions detected" in H1 2022 (11,8 % of all notifications). The annual report provides a technical description of the notification category, you can read that report here.

"Vulnerable external SSL connections" were slightly less prevalent in Q1 2022 (14,6% of all notifications) compared to Q1 2021 (10,9% of all notifications). We will provide a technical explanation of this notification category in a subsequent section of this report.

"Invalid SSL certificates from internal servers" increased slightly in H1 2021 (8,9% of all notifications) compared to 2021.

The notification "External DNS server" decreases to 10,3%.

"Vulnerable external SSL connections" were slightly less prevalent in H1 2022 (10,1% of all notifications) compared to 2021.

The 'Invalid SSL certificate from external server' drops to 9.1%.

We will provide a technical explanation of these notification categories - "Vulnerable external SSL connection" and "Expired SSL certificate from external server" - in a subsequent section of this report.

* The following categories had under 1% (Other).

Lateral movement using SMB admin shares
Selective Port Scan
Soon to expire SSL certificate from internal server
Expired SSL certificate from internal server
OT unknown function codes
Anomaly - Unexpected Service
External SMTP e-mail server
Port scan detected
SSH Interesting Hostname Login
RDP Outgoing Connection
Expired SSL certificate from external server
Tor middle node communication
Exfiltration of many files
HTTP SQL injection detected
Blacklist match file

Anomaly - Data Transfer
External POP3 e-mail server
SSH Failed Attempts
Soon to expire SSL certificate from external server
DNS Multiple Domain Not Found
Anomaly - Out of hours
Tor exit node connection
Misconfigured HTTP basic auth client
Anomaly - Unexpected Service and Port
Blacklist match certificate
P2P traffic patterns
Kerberos Longlived Ticket
HTTP SQL injection victim detected
Secure com password guessing attempts detected
DNS over HTTPs (DOH) usage

HTTP crawler detected
SMB Suspicious File Renaming
Event log clearing or forced reboot using RPC
DNS Tunneling
Impossible travel detected
Login from an unprecedented country
Anomaly - Unusual Context
Global port scan
Reverse SSH
Large amount of files downloaded
FTP brute force login detected
DarkNet or Tor activity detected
Local blacklisted executable detected
Not yet valid SSL certificate from internal server
HTTP Authentication Bruteforce

Lateral movement and execution
Too many failed login attempts for user
Too many failed login attempts from IP
Blacklist match SSH
BitTorrent port usage
Crypto Currencies Mining Pool Activity
Large amount of mail attachment sent
Login from an unexpected country
NTLM User Password Bruteforce
Failed login attempt from an unprecedented country
Blacklist match domain
SMB Ransomware filename detected

Technical deep dive:
Vulnerable External SSL Connection

by Henrik Falkenthros
Senior IT Security Engineer
https://www.linkedin.com/in/henrikfalkenthros

Threat

Muninn is designed to identify vulnerable encryption protocols to protect information that traverses networks and therefore looks for vulnerable (unencrypted) SSL connections in the network traffic, helping security managers proactively mitigate vulnerabilities in the network.

Exploitation

The SSL and TLS protocols have always been the target of many kinds of cyber-attacks. The threat actor will often conduct a "Man in The Middle-attack" to access the content of the encrypted communication. Bypassing the protocols, downgrading, stripping them, or reading from memory are also common attack vectors.

There are quite a few options to attack vulnerable SSL connections. A simple Google query using the keywords FREAK, DROWN, SWEET32, HeartBleed, POODLE, BREACH, CRIME will yield many results.

Information on how to conduct Man In the Middle Attacks is easy to find online, and github has this repository.

As in most cases, the hacker will target the weakest link to break into the company (and of course, the target must be of interest and value).

Let's use https://samplelabserver.com as a fictitious example of the target.

On a kali box, you can check the webserver that you want to communicate with by running sslscan:

```
sirhenry@192.168.2.126 :~/Documents/Godfathers/ip-map/ipranges
$sslscan samplelabserver.com
Version: 2.0.7
OpenSSL 1.1.1f  31 Mar 2020

Connected to 186.2.169.7

Testing SSL server samplelabserver.com on port 443 using SNI name  samplelabserver.com

  SSL/TLS Protocols:
SSLv2     disabled
SSLv3     disabled
TLSv1.0   enabled
TLSv1.1   enabled
TLSv1.2   enabled
TLSv1.3   disabled

  TLS Fallback SCSV:
Server supports TLS Fallback SCSV

  TLS renegotiation:
Secure session renegotiation supported

  TLS Compression:
OpenSSL version does not support compression
Rebuild with zlib1g-dev package for zlib support

  Heartbleed:
TLSv1.2 not vulnerable to heartbleed
TLSv1.1 not vulnerable to heartbleed
TLSv1.0 not vulnerable to heartbleed

  Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256   Curve P-384 DHE 384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384   Curve P-384 DHE 384
Accepted  TLSv1.2  128 bits  DHE-RSA-AES128-GCM-SHA256     DHE 2048 bits
Accepted  TLSv1.2  256 bits  DHE-RSA-AES256-GCM-SHA384     DHE 2048 bits
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256       Curve P-384 DHE 384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384       Curve P-384 DHE 384
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA          Curve P-384 DHE 384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA          Curve P-384 DHE 384
Accepted  TLSv1.2  128 bits  DHE-RSA-AES128-SHA256         DHE 2048 bits
Accepted  TLSv1.2  128 bits  DHE-RSA-AES128-SHA            DHE 2048 bits
Accepted  TLSv1.2  256 bits  DHE-RSA-AES256-SHA256         DHE 2048 bits
Accepted  TLSv1.2  256 bits  DHE-RSA-AES256-SHA            DHE 2048 bits
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA
Accepted  TLSv1.2  256 bits  AES256-SHA
Accepted  TLSv1.2  112 bits  TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.2  112 bits  TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.2  112 bits  TLS_RSA_WITH_3DES_EDE_CBC_SHA
Preferred TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA          Curve P-384 DHE 384
Accepted  TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA          Curve P-384 DHE 384
Accepted  TLSv1.1  128 bits  DHE-RSA-AES128-SHA            DHE 2048 bits
Accepted  TLSv1.1  256 bits  DHE-RSA-AES256-SHA            DHE 2048 bits
Accepted  TLSv1.1  128 bits  AES128-SHA
Accepted  TLSv1.1  256 bits  AES256-SHA
Accepted  TLSv1.1  112 bits  TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.1  112 bits  TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.1  112 bits  TLS_RSA_WITH_3DES_EDE_CBC_SHA
Preferred TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA          Curve P-384 DHE 384
Accepted  TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA          Curve P-384 DHE 384
Accepted  TLSv1.0  128 bits  DHE-RSA-AES128-SHA            DHE 2048 bits
Accepted  TLSv1.0  256 bits  DHE-RSA-AES256-SHA            DHE 2048 bits
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  256 bits  AES256-SHA
Accepted  TLSv1.0  112 bits  TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.0  112 bits  TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Accepted  TLSv1.0  112 bits  TLS_RSA_WITH_3DES_EDE_CBC_SHA

  Server Key Exchange Group(s):
TLSv1.2  192 bits  secp384r1 (NIST P-384)

  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  [redacted]
Altnames: [redacted]
Issuer:   [redacted]

Not valid before: Feb 10 00:00:00 2018 GMT
Not valid after:  Feb 10 23:59:59 2019 GMT


sirhenry@192.168.2.126 :~/Documents/Godfathers/ip-map/ipranges
```

Listed on the next page are the eight important sections you should have a closer look at.

1. SSL/TLS protocols

SSLv2 and v3 are disabled meaning that the communication between the client and the webserver will not be able to use this weak protocol for network encryption. So far so good, but is worrying that TLSv1.0 and v1.1 are supported since they are also known to be weak and vulnerable. What really should catch your attention is the fact that TLSv3 is disabled, because this is what is generally considered the de-facto standard for secure network encryption. See the difference between TLSv1.2 and TLSv1.3 here.

2. TLS fallback SCSV

The TLS Signaling Cipher Suite Value (SCSV) protects against downgrade attacks. If enabled, the server makes sure that the strongest protocol that both client and server supports is used. In this case every thing is good since it is activated.

3. TLS renegotiation

The webserver supports secure fallback, which means that authentication details can be added to the current connection if need be.

4. TLS Compression

As part of the TLS Handshake, it includes features to negotiate data compression method
In the screenshot, sslscan does not check for this, so use nmap to verify:

```
nmap -sV --script ssl-enum-ciphers -p 443 samplelabserver.com
```

in the output, nmap says no compression is used. Note, this script also checks for known vulnerabilities such as SWEET32.

5. Heartbleed

The webserver is not vulnerable to heartbleed, a famous memory leak flaw in OpenSSL.
Verify using nmap:

```
nmap -d --script ssl-heartbleed --script-args vulns.showall -sV samplelabserver.com
```

The output 'ssl-heartbleed: NOT VULNERABLE' confirms that the webserver is not vulnerable.

6. Supported Server Ciphers

Cipher suites are a number of algorithms to secure SSL or TLS network connections. The cipher suites usually include: a key exchange algorithm, a bulk encryption algorithm and a message authentication code (MAC) algorithm. The ciphers marked with green are secure suites and are preferrable to use. The ones in white and orange are weak and should never be used. For more information about cipher suites, consult IANA homepage.

7. Server Key Exchange Group

The P-384 is an elliptic curve that provides a 192-bit security level and is used in the computation of digital signatures and key-agreement protocols.

8. SSL Certificate

Signature Algorithm: sha256WithRSAEncryption. The Signature Algorithm represents the hash algorithm used to sign the SSL certificate. RSA Key Strength: 2048. This is sufficient, and also note that the certificate is not valid anymore.

So going through the eight steps, there are plenty of reasons not to connect to this website, the main reasons being the lack of a valid certificate, the support of TLSv1.0 and the non-support of TLSv3.

From a hacker's point of view, the website supports the weak TLSv1.0 with 112 bits length and support of triple DES and CBC, is clearly a possible attack vector.

A hacker would verify this by running the following nmap command:

```
nmap -Pn --script ssl-enum-ciphers -p 443 samplelabserver.com
```

The result shows '64-bit block cipher 3DES vulnerable to SWEET32 attack'

So now the hacker will perform a SWEET32 attack by collecting a lot of https traffic between the target and the webserver. Hereafter, a cracking tool must be executed.

See this repository for a practical DIY reduced version of the SWEET32 attack.

Severity

According to for instance the NVD database, alone in 2022, 10+ new CVE's are recorded concerning SSL/TLS known breaches. Since 1999, the number is over 3,500. The rating often starts from CVSSv2 around 5.0 (medium) to 10.0 (High), and it may seem obvious since it is concerning breaking network securing protocols.

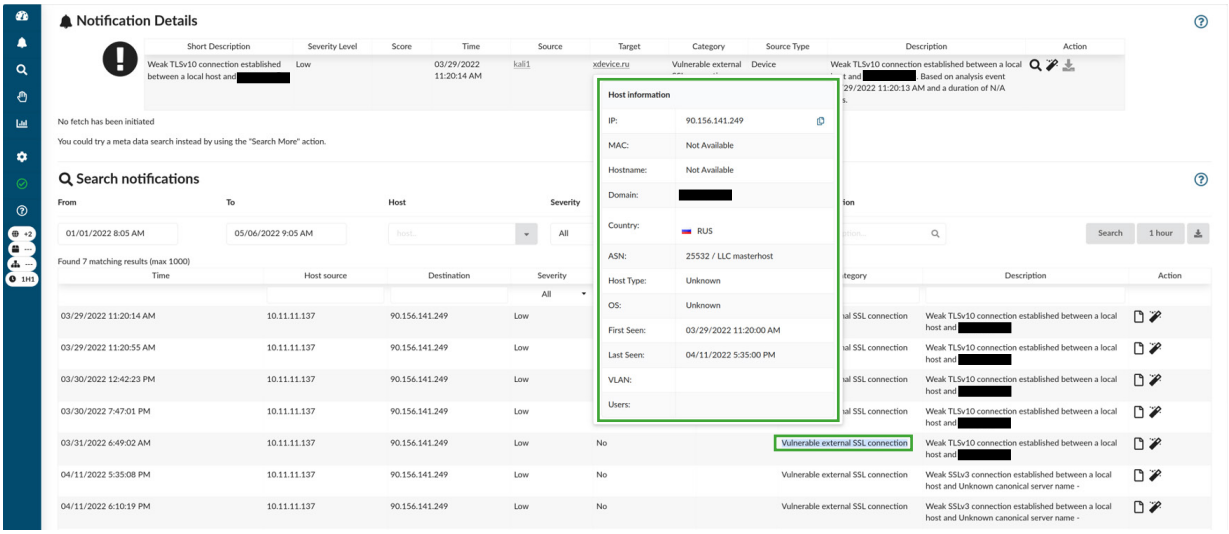
What we see in Muninn

The Muninn sensor is designed to catch communication with vulnerable SSL connections to webservers inside the LAN or to external. By analysing the initial ethernet frames, we can see from the handshakes what type of network encryption is negotiated between client and webserver and thus to be used in the HTTPS session.

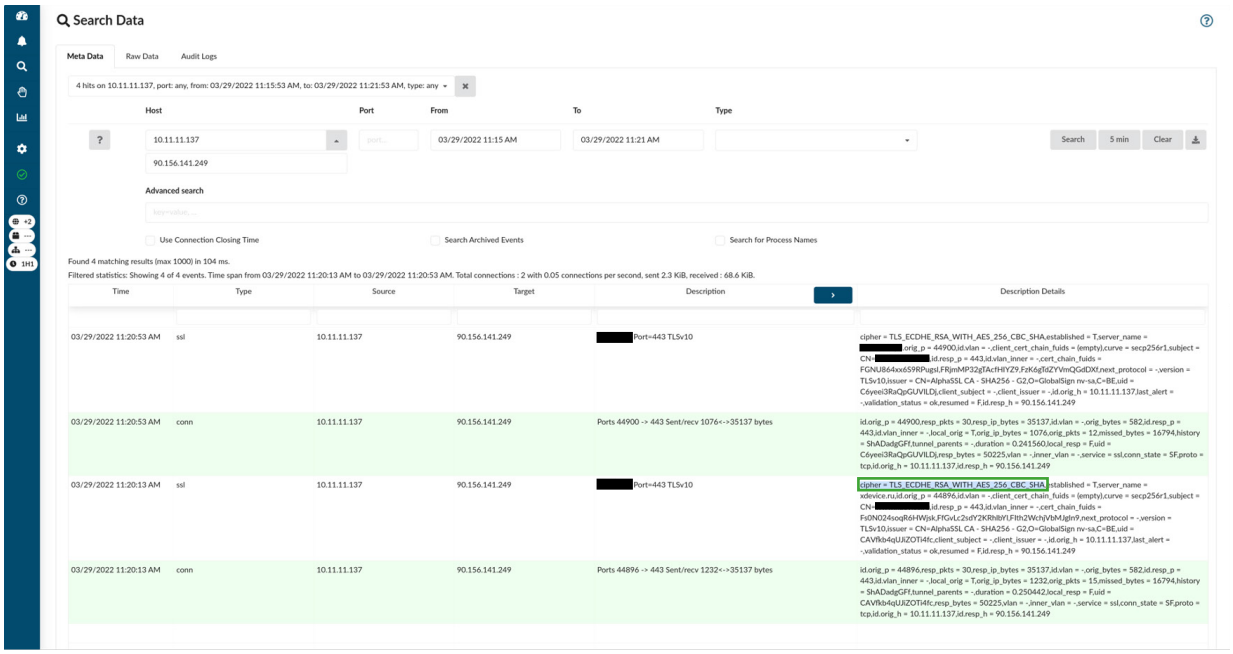
The initial packets are used to obtain information of an encrypted communication session. It allows extraction of interesting data such as an HTTP URL, DNS hostname/address etc. The TLS handshake is composed of several messages that contain interesting, unencrypted metadata like cipher suites, TLS versions and the client's public key length. Learn more about TLS here.

On the next page is an example of a notification that has been triggered because of the usage of TLSv1.0.

> The SSL and TLS protocols have always been the target of many kinds of cyber-attacks. The threat actor will often conduct a "Man in The Middle-attack" to access the content of the encrypted communication. Bypassing the protocols, downgrading, stripping them, or reading from memory are also common attack vectors.



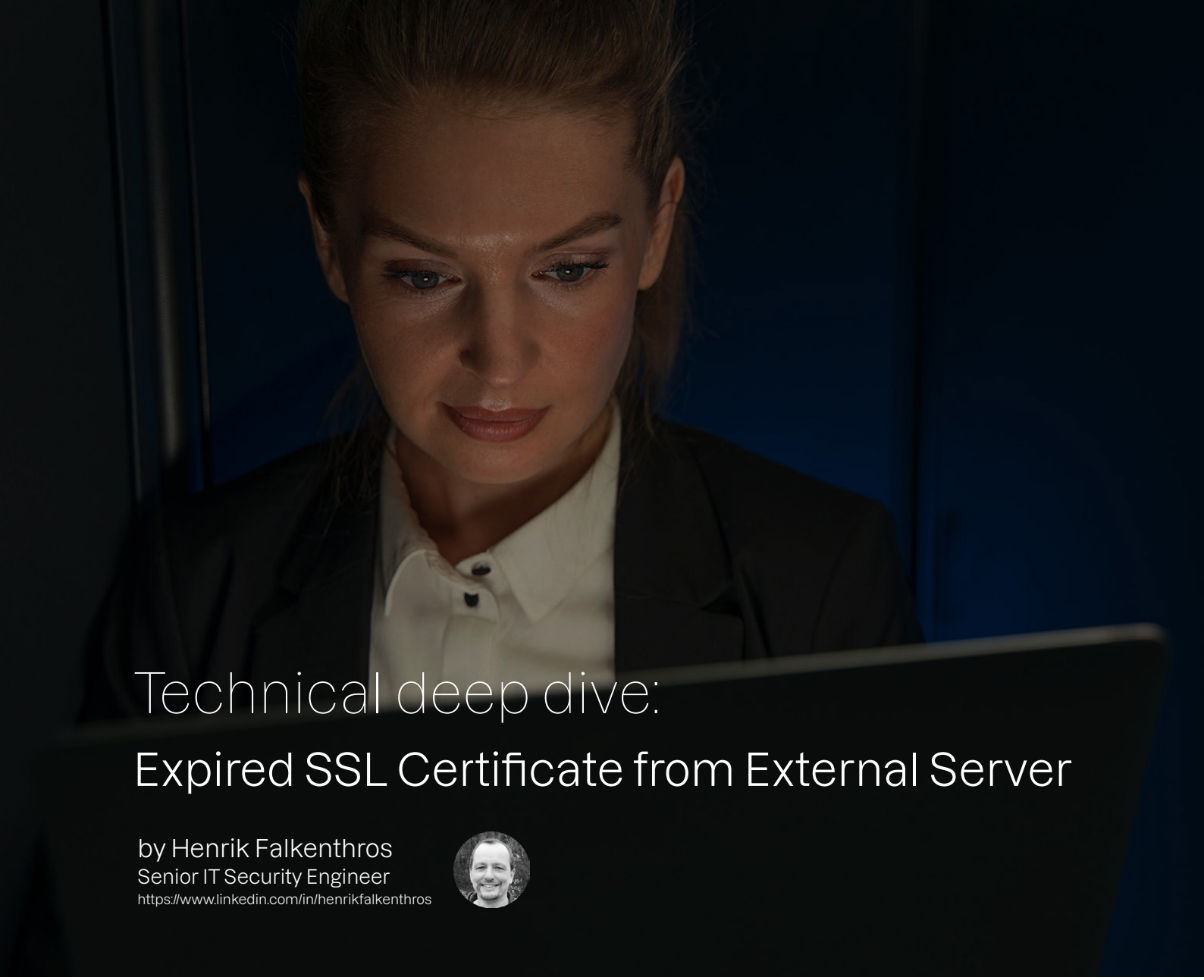
Searching the metadata reveals the cipher suite used in the communication, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA:



Counter-measurements

Support only TLSv1.3 and above.

> The Muninn sensor is designed to catch communication with vulnerable SSL connections to webservers inside the LAN or to external. By analysing the initial ethernet frames, we can see from the handshakes what type of network encryption is negotiated between client and webserver and thus to be used in the HTTPS session.



Technical deep dive:

Expired SSL Certificate from External Server

by Henrik Falkenthros
Senior IT Security Engineer
https://www.linkedin.com/in/henrikfalkenthros

Threat

A certificate enables two features, namely Authentication and Encryption. So, when a certificate expires, the user no longer has the guarantee of the server being authentic and belonging to the company the certificate was issued to by the CA.

This is because the public CA only signs a certificate if the owner is verified. A rogue website will not have the private key of the original certificate, as that is solely owned by the original webserver.

As a user you'll encounter a number of warnings if the certificate is expired, and thus not trustworthy to communicate with. This is of cause worrying, but the expired certificate still provides same encryption as before the expiration date.

For public facing websites, these warnings will create uncertainty for the users and potentially scare them away from doing business online.

Exploitation

Hackers are known to have lists of a wide range of companies that they monitor for certification expiration dates.

As part of their OSINT information gathering, they have created a fake website using the same graphic identity similar to the target and prepared the list of users to attack. And of course, purchased a plausible domain name like 'verification-company.com' and a valid certificate.

Once the certificate has expired the phishing attacker starts by sending out e-mails to the company's customer saying "... due to security issues, you need to verify your account on our new verification site at https://www.verification-company.com".

A very few users will actually investigate the certificate details and will probably enter credentials to this rouge website.

As an example, do your google dorking (webshops, retailers, fashion) and create a list with companies you'd like to attack e.g. 'top-50-shops.lst'.

From github or any other repository, download a script that checks for certificate expiration (such as this one) and do a one-liner in kali:

```
Clear; while read -r line; do sudo  timeout 5 python3 check_certificates.py $line | awk '{print
$1 "\t" "expires in "$5" "$6}'; done < top-50-shops.lst
```

This provides the following information;

```
Alibaba.com             expires in 261 days     hepsiburada.com         expires in 316 days
aliexpress.com          expires in 353 days     Ikea.com                expires in 107 days
allegro.pl              expires in 72 days      Inditex.com             expires in 317 days
amazon.ca               expires in 321 days     jd.com                  expires in 137 days
Amazon.co.jp            expires in 82 days      johnlewis.com           expires in 27 days
Amazon.co.uk            expires in 82 days      kakaku.com              expires in 124 days
Amazon.com              expires in 82 days      leboncoin.fr            expires in 267 days
amazon.com.br           expires in 310 days     lego.com                expires in 83 days
amazon.com.mx           expires in 319 days     mercadolibre.com.ar     expires in 247 days
Amazon.de               expires in 82 days      mercadolibre.com.mx     expires in 257 days
amazon.es               expires in 330 days     mercadolivre.com.br     expires in 248 days
amazon.fr               expires in 319 days     mercari.com             expires in 209 days
Amazon.in               expires in 325 days     olx.com.br              expires in 316 days
amazon.it               expires in 328 days     olx.pl                  expires in 231 days
americanas.com.br       expires in 239 days     ottogroup.com           expires in 377 days
Apple.com               expires in 331 days     ozon.ru                 expires in 279 days
bestbuy.com             expires in 204 days     pinduoduo.com           expires in 275 days
Bol.com                 expires in 152 days     rakuten.co.jp           expires in 142 days
Carrefour.com           expires in 171 days     sahibinden.com          expires in 293 days
Ceconomy.de             expires in 83 days      Shop.com                expires in 39 days
costco.com              expires in 310 days     shopee.co.id            expires in 127 days
craigslist.org          expires in 240 days     shopping.yahoo.co.jp    expires in 345 days
e.leclerc               expires in 114 days     taobao.com              expires in 110 days
ebay-kleinanzeigen.de   expires in 111 days     Target.com              expires in 123 days
Ebay.co.uk              expires in 213 days     Tesco.com               expires in 50 days
Ebay.com                expires in 213 days     ticketmaster.com        expires in 353 days
ebay.de                 expires in 213 days     tokopedia.com           expires in 82 days
ecco.com                expires in 158 days     trendyol.com            expires in 327 days
etsy.com                expires in 262 days     Veepee.fr               expires in 199 days
Flipkart.com            expires in 23 days      walmart.com             expires in 305 days
groupe-casino.fr        expires in 128 days     wayfair.com             expires in 88 days
Groupon.com             expires in 225 days     Zalando.com             expires in 140 days
```

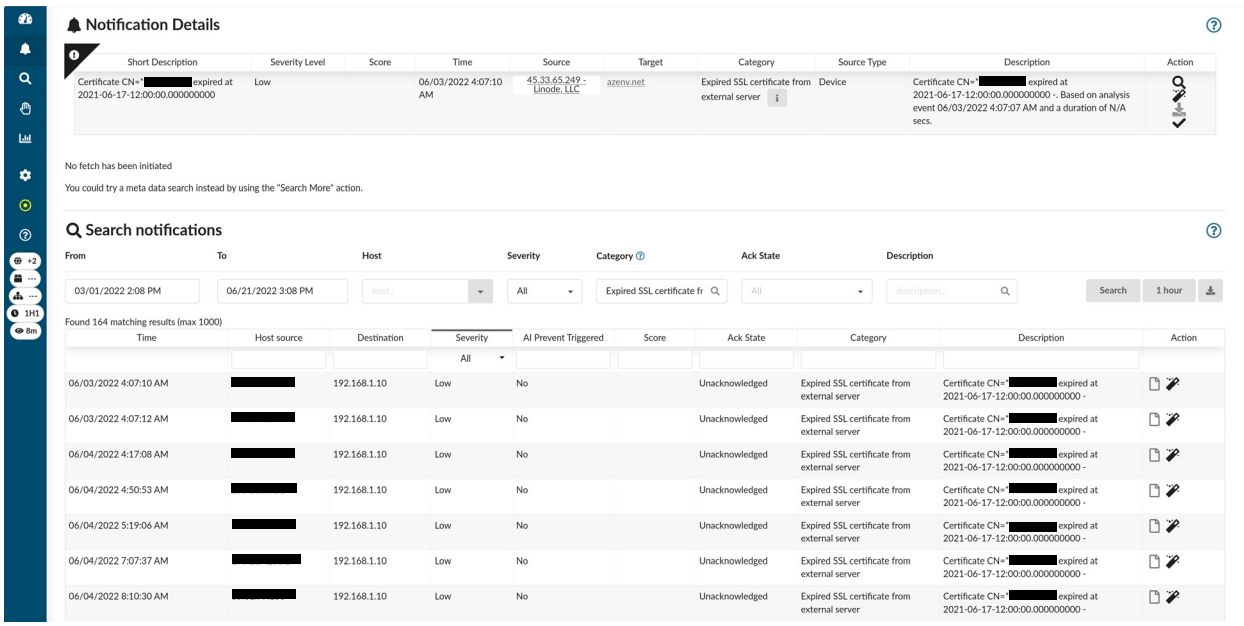
So, all you need to do is to find e-mail accounts for say LEGO.com and compose a thrustworthy phishing e-mail.

There are many ways to get users e-mail addresses tied to a company with a webshop indicating a commercial relationship. A good place to start is diving into blogs dealing with the company's product or services. Often people give away e-mail addresses and full names.

Social medias like Facebook, Twitter, Linkedin contain lots of information, too, useful when chasing product / service interests. If you do not want to do it yourself, you can always buy a darknet service for a relatively small amount of money.

What we see from Muninn

The Muninn sensor reacts on the value 'NotValidAfter' and create the following notification:



Notification Details

Short Description	Severity Level	Score	Time	Source	Target	Category	Source Type	Description	Action
Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000	Low		06/03/2022 4:07:10 AM	45.33.65.249 - Linode, LLC	azenv.net	Expired SSL certificate from external server	Device	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -. Based on analysis event 06/03/2022 4:07:07 AM and a duration of N/A secs.	

No fetch has been initiated

You could try a meta data search instead by using the "Search More" action.

Search notifications

From: 03/01/2022 2:08 PM | To: 06/21/2022 3:08 PM | Host: host.. | Severity: All | Category: Expired SSL certificate fr | Ack State: All | Description: description... | Search | 1 hour

Found 164 matching results (max 1000)

Time	Host source	Destination	Severity	AI Prevent Triggered	Score	Ack State	Category	Description	Action
			All						
06/03/2022 4:07:10 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/03/2022 4:07:12 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/04/2022 4:17:08 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/04/2022 4:50:53 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/04/2022 5:19:06 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/04/2022 7:07:37 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	
06/04/2022 8:10:30 AM	[redacted]	192.168.1.10	Low	No		Unacknowledged	Expired SSL certificate from external server	Certificate CN=[redacted] expired at 2021-06-17-12:00:00.000000000 -	

Counter-measurements

- Do not communicate with servers/services that cannot present a valid certificate
- If the website is malicious, permanently block it in the firewall
- If the website was legitimate but has an expired SSL certificate, temporarily block the domain in your firewall and inform your external partner that they are using expired certificates.

> As part of their OSINT information gathering, hackers create fake websites using the same graphic identity similar to the target, and prepared the list of users to attack. And of course, they purchase a plausible domain name like 'verification-company.com' and a valid certificate.

External Links & References

https://sweet32.info/

https://heartbleed.com/

https://www.exploit-db.com/

https://www.keylength.com/en/4/

https://nmap.org/book/man-nse.html

https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art060

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-186-draft.pdf

https://www.researchgate.net/publication/322056853_Speeding_up_Elliptic_Curve_Cryptography_on_the_P-384_Curve

https://nvd.nist.gov/vuln/search/results?results_type=overview&query=ssl&search_type=all&form_type=Basic&isCpeNameSearch=false&orderBy=publishDate&orderDir=desc

https://github.com/azeemba/sour16/blob/master/ - practical kali demo of cracking encryption

https://ciphersuite.info/search/?security=all - secure cipher suites

https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

https://docs.microsoft.com/en-us/windows/win32/seccertenroll/about-x-509-public-key-certificates

https://gbhackers.com/latest-google-dorks-list/

Abbreviations

CA - Certificate Authority

CVSS - Common Vulnerability Scoring System

DIY – Do It Yourself

NVD – National Vulnerability Database

OSINT – OpenSource INTelligense

TLS - Transport Layer Security

RSA – Rivest Shamir Adleman

SHA - Secure Hash Algorithm

X509 - format of public key certificates

NVD - National Vulnerability Database

About Muninn

AI powered cyber security. Muninn understands and oversees your entire digital infrastructure, prevents data leakage, and protects critical business assets from disruptive cyber-attacks. We maximize business continuity.

Analyzes the entire network and AI calculates baseline traffic pattern. Discovers abnormal and suspicious behavior and reports real threats.

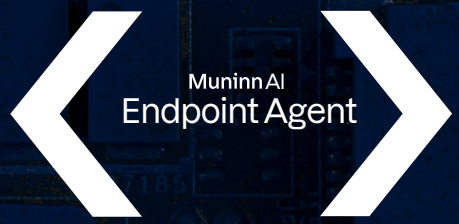
→ www.muninn.ai/detect

Stops data leakage and ransomware in milliseconds. Autonomously neutralizes sophisticated cyber threats, enabling business continuity.

→ www.muninn.ai/prevent

AI technology to stop phishing-emails before reaching the inbox in cloud environments such as Microsoft365 and Google Apps.

→ www.muninn.ai/muninn-ai-anti-phishing



Detects and responds to cyber threats targeting endpoints. Covers branch offices and remote workers to extend the coverage of Muninn AI Detect.

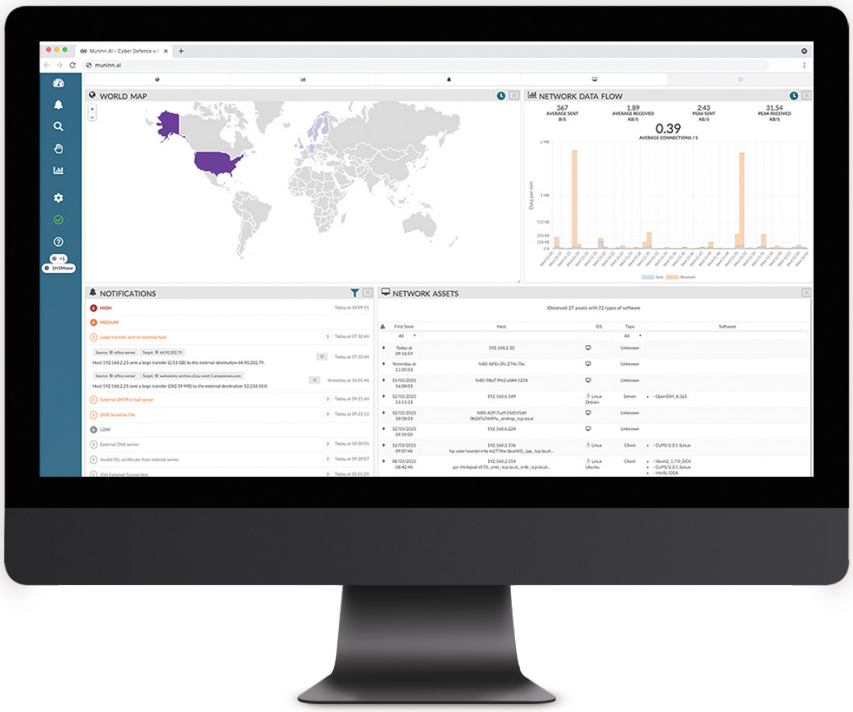
→ www.muninn.ai/muninn-ai-endpoint-agent

Muninn was founded by computer scientists from the Massachusetts Institute of Technology (M.I.T), engineers, and cyber security experts with government intelligence backgrounds.

The team's dedication to protecting critical national infrastructure is the foundation of Muninn, which as a commercial product now protects companies and institutions across Europe and North America.

sales@muninn.ai +45 70 60 59 08 www.muninn.ai Muninn ApS, Lyngby Hovedgade 94, st, 2800 Kongens Lyngby, Denmark

Try Muninn



Free

Scan the code to request a trial

www.muninn.ai/freetrial

info@muninn.ai

+45 70 60 59 08