black hat ASIA 2022

VXRL

Backdoor Investigation and Incident Response: From Zero to Profit

VXRL - Anthony Lai, Alan Ho, Ken Wong, Zetta Ke

Speaker Bio: Anthony Lai

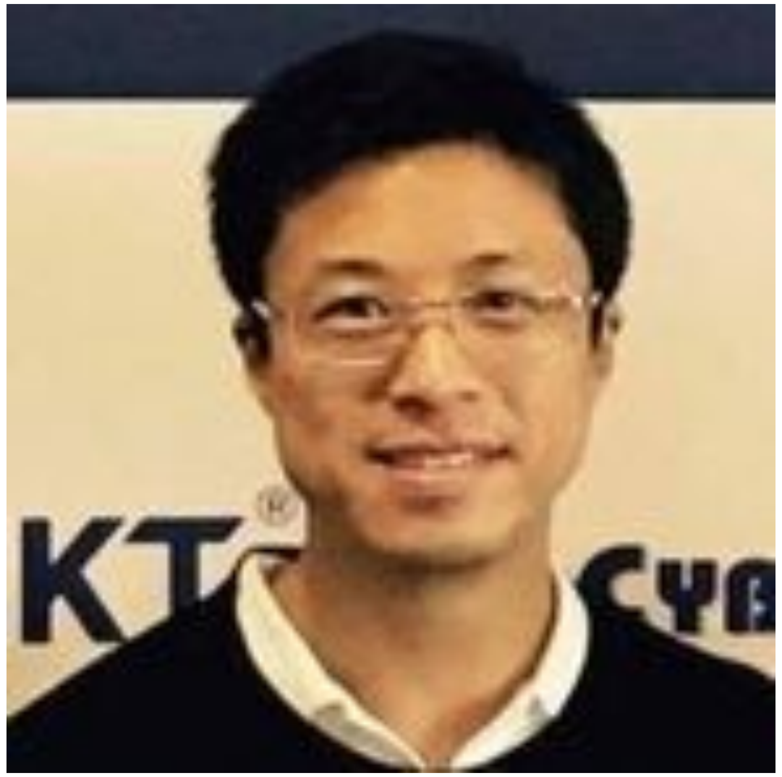
- VXRL and VXCON Chairperson
- Enjoy hacking since 2003
- Focus on threat analysis, incident response, and red/blue team testing
- Overseas mentor @ Best of the Best (BoB) Korea Hacker Educational Program
- Blackhat Asia and HITB CFP Reviewer
- AVTokyo, Blackhat USA, DEFCON, Codegate, DFRWS, HITB, HITCON and Secuinside speakers
- Computer Science PhD Candidate @ HKUST

Speaker Bio: Ken Wong

- Computer Science PhD student @ HKUST
- VXRL researcher
- Interested in software security, binary analysis and machine learning
- Presented in different conferences, e.g., HITCON and VXCON
- Founding member of Black Bauhinia CTF team

Speaker Bio: Alan Ho

- VXRL and VXCON co-founder
- Focus on penetration test, red / blue team test, incident response, training and security operation planning
- OSCP, SANS GWAPT, GCIH Holder
- Presented in different conferences, e.g., AVTokyo, DEFCON China and USA villages, SANS DFIR, DFRWS, HITCON
- Provide seminars and training to schools and community



Speaker Bio: Zetta Ke

- Assistant Professor in School of Computing and Information Systems in Singapore Management University
- VXRL and VXCON co-founder
- Interested in web application security, blockchain security, and economics of security
- Received PhD degree from HKUST in 2018
- Presented in different conferences, e.g., AVTokyo, DEFCON Village, HITCON
- Founding member of Black Bauhinia CTF team

Introduction

Why we still need a talk on Backdoor Investigation and Incident Response?

- Popular malware analysis tools like VirusTotal is already powerful and easy to use.
- The high-level methodology in incident response existing incident response frameworks is already comprehensive.
- But they failed to detect and identify highly stealthy malware backdoor and attack origin.

Case Background

- Our team is responsible for incident handling for a company with online gaming business in APAC, and with development team in Hong Kong.
- Due to COVID19, staff worked from home. They mostly used SSL VPN to connect to company network.
- One day, network team reported SSL VPN connection and unauthorized modification of firewall rules during unusual time in midnight.
- Our team started to investigate the case.
- After a month, another alert was found from IIS.

Investigation

Interviewed the network engineers and system administrator:

- No tasks were assigned to them during that day in the midnight.
- Attackers logon their Windows Active Directory account to access Firewall Admin UI in their desktops.
- The engineers had no ideas of why their accounts were compromised.
- Firewall Admin Account was not using 2FA.
- SSL VPN logons were not using 2FA.

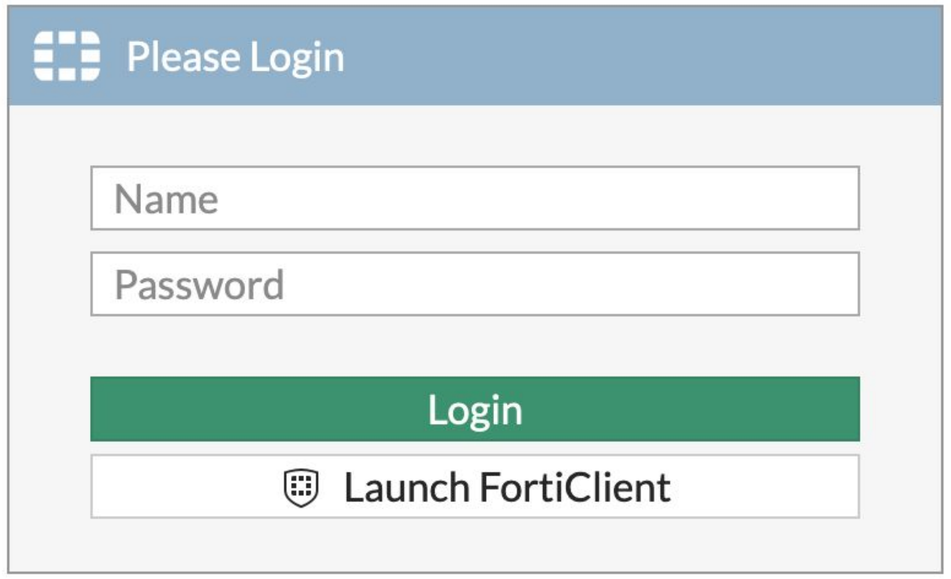
Investigation

Check against Splunk Logs

- The SSL VPN connection was made from a non-Hong Kong IP Address.

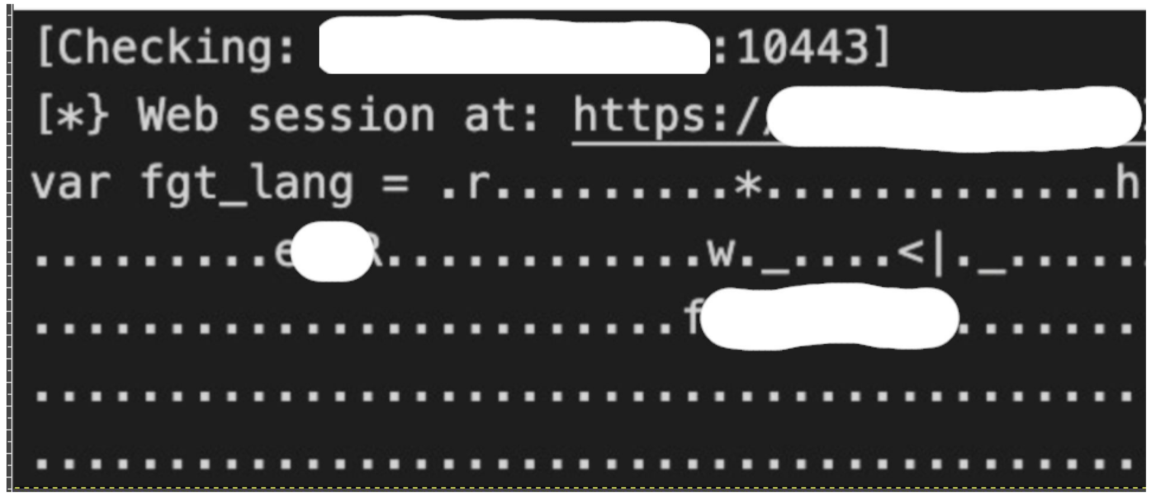
Check SSL VPN Device

- Fortigate SSL VPN was used.
- WebUI https://xx.xx.xx.xx:10443 was used.



SSL VPN Exploit

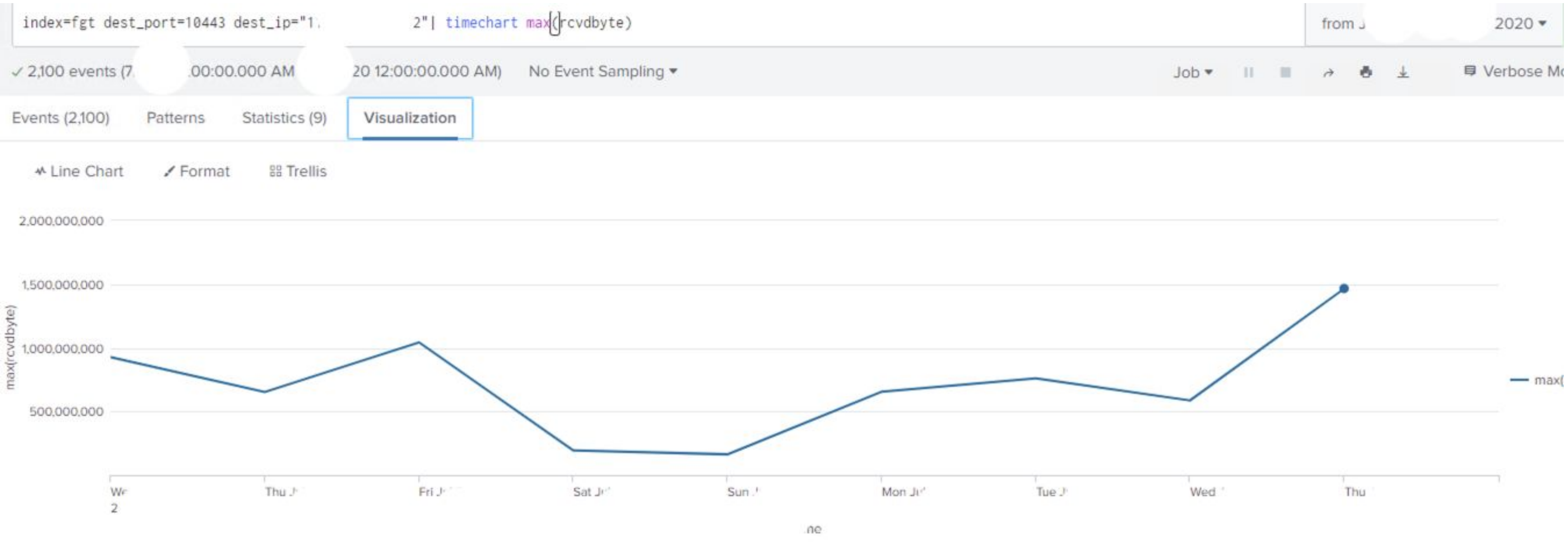
- Fortigate SSL VPN was vulnerable to credential leakage (CVE-2018-13379: Pre-auth arbitrary file reading) (https://devco.re/blog/2019/08/09/attacking-ssl-vpn-part-2-breaking-the-Fortigate-ssl-vpn/).
- Our team started the Red Team Test, and successfully retrieved all the credentials of SSL VPN accounts.



```
[Checking: [redacted]:10443]
[*} Web session at: https:/[redacted]
var fgt_lang = .r.........*.............h
.........e[redacted]R............w._.....<|._.....
........................f[redacted].......
.................................
.................................
```

Verification from Splunk

- Blue Team Techniques to check the Splunk Logs.
- Checking the logs of traffic to SSL VPN.
- Spikes of “bytes received” from the suspicious address.
- Normally establishing SSL VPN connection will produce steady traffic, However when running the exploit, since attacker will receive credentials, the byte received will be much larger than usual.

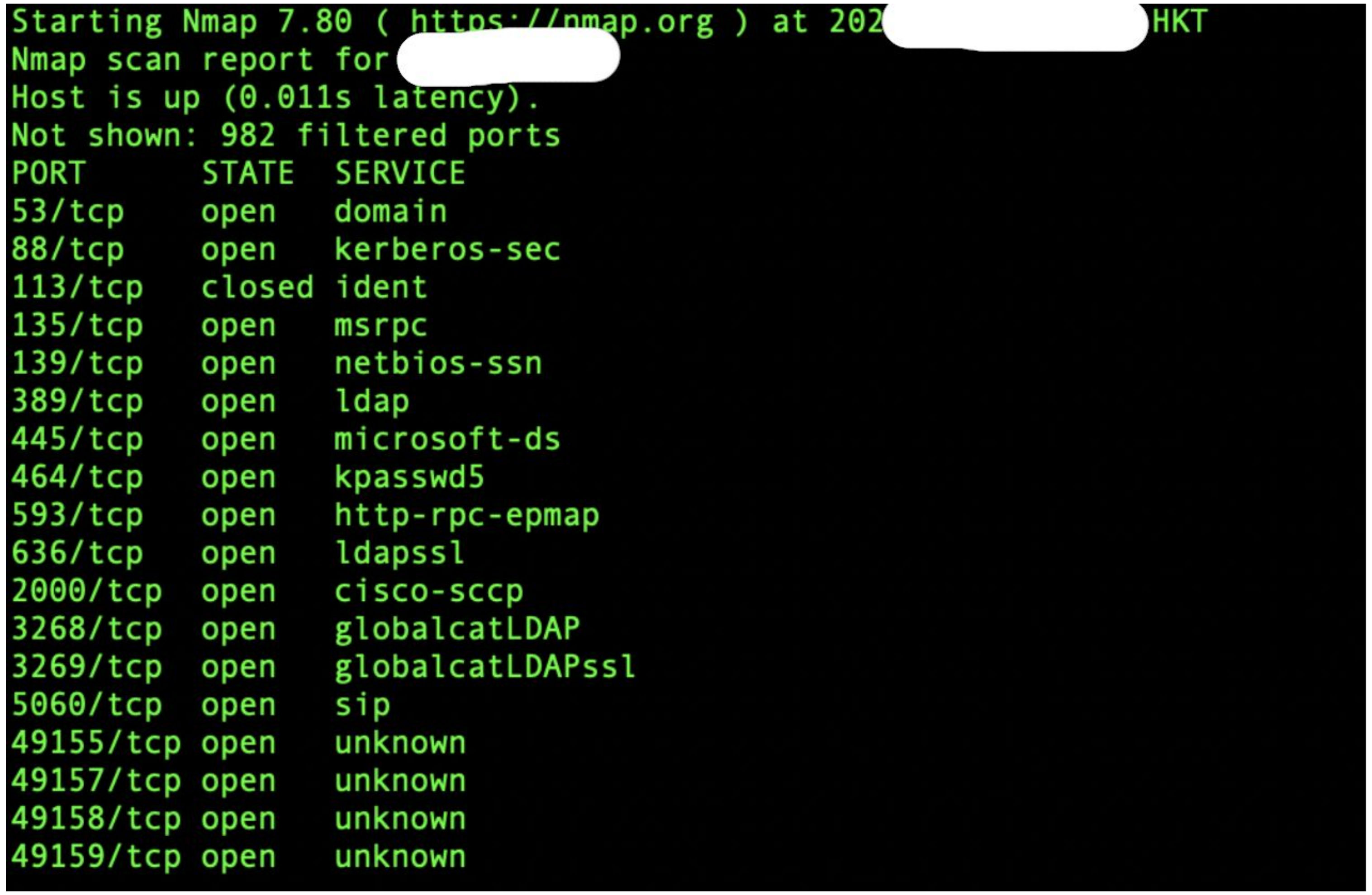


```
index=fgt dest_port=10443 dest_ip="1.          2"| timechart max(rcvdbyte)
```

Next Step of Exploit POC

- During the previous exploitation, we successfully retrieved the SSL VPN credentials
- Next step, we used the accounts to login the company intranet by SSL VPN
- After a bit of network discovery, our team found the IP Addresses of the Active Directory.

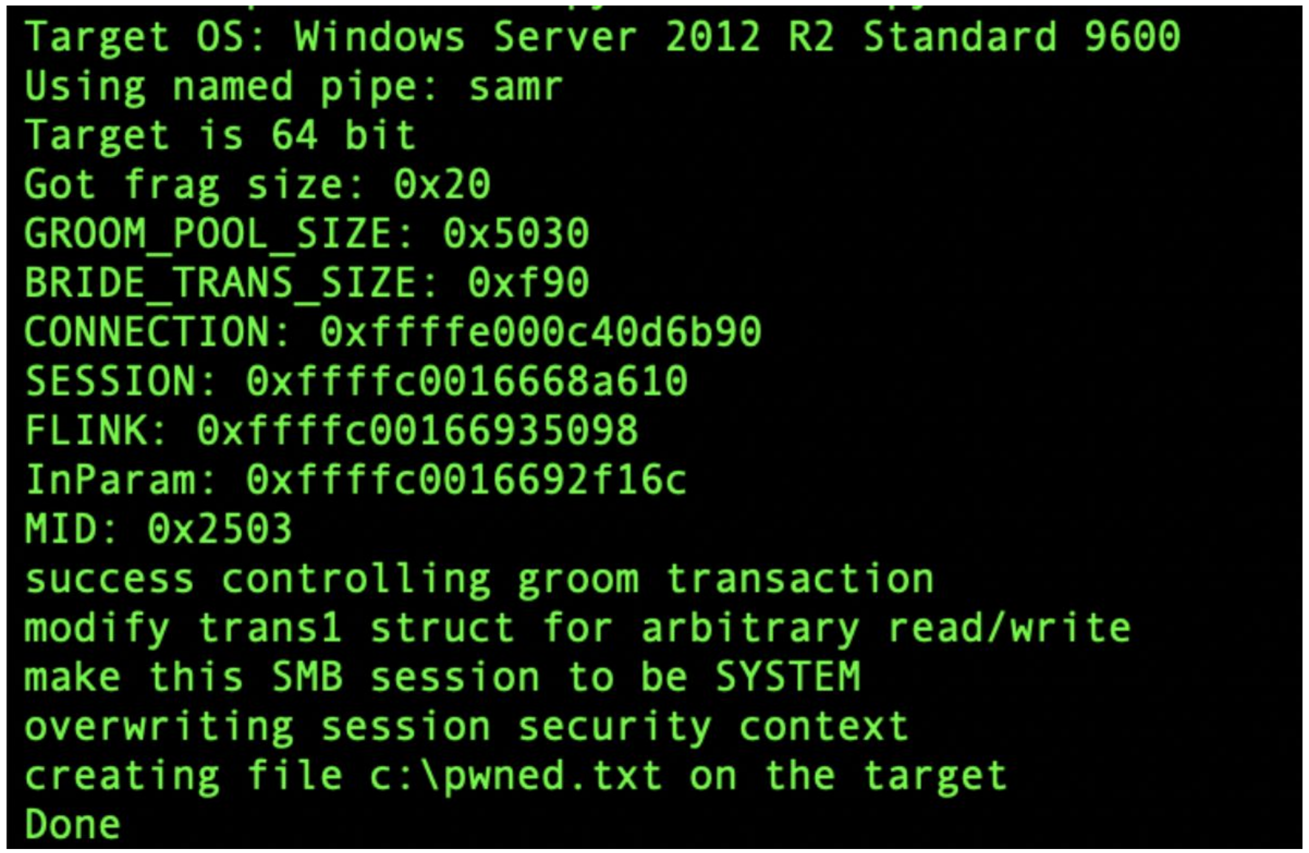
Port scanning on Active Directory



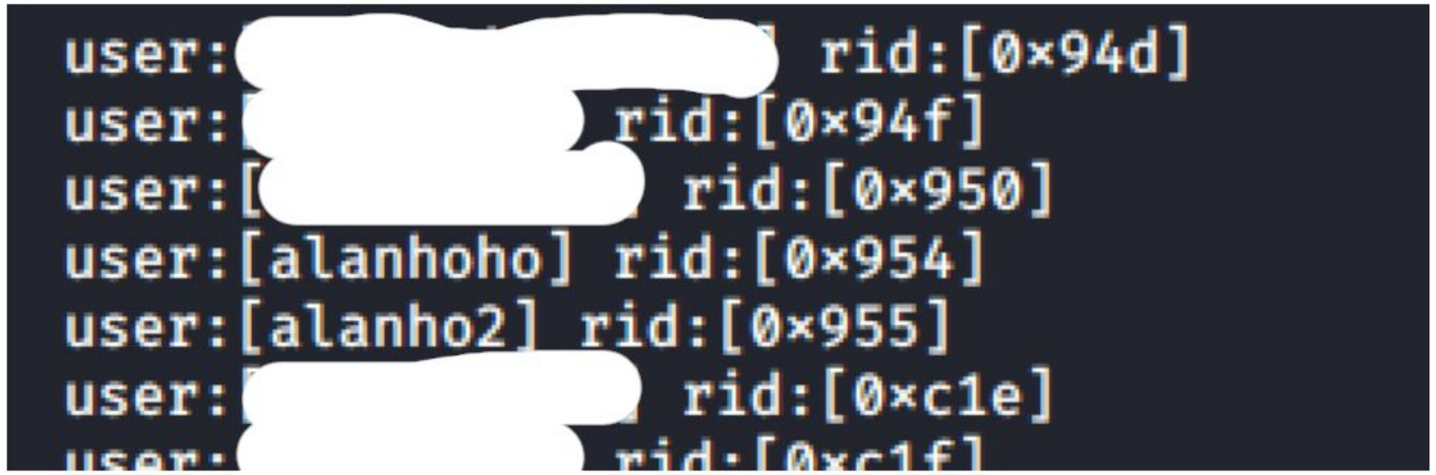
```
Starting Nmap 7.80 ( https://nmap.org ) at 202[redacted]HKT
Nmap scan report for [redacted]
Host is up (0.011s latency).
Not shown: 982 filtered ports
PORT      STATE  SERVICE
53/tcp    open   domain
88/tcp    open   kerberos-sec
113/tcp   closed ident
135/tcp   open   msrpc
139/tcp   open   netbios-ssn
389/tcp   open   ldap
445/tcp   open   microsoft-ds
464/tcp   open   kpasswd5
593/tcp   open   http-rpc-epmap
636/tcp   open   ldapssl
2000/tcp  open   cisco-sccp
3268/tcp  open   globalcatLDAP
3269/tcp  open   globalcatLDAPssl
5060/tcp  open   sip
49155/tcp open   unknown
49157/tcp open   unknown
49158/tcp open   unknown
49159/tcp open   unknown
```

Eternalblue

- Our team tried Eternalblue exploit written in Python and uploaded a POC file to the AD server and created accounts



```
Target OS: Windows Server 2012 R2 Standard 9600
Using named pipe: samr
Target is 64 bit
Got frag size: 0x20
GROOM_POOL_SIZE: 0x5030
BRIDE_TRANS_SIZE: 0xf90
CONNECTION: 0xffffe000c40d6b90
SESSION: 0xffffc0016668a610
FLINK: 0xffffc00166935098
InParam: 0xffffc0016692f16c
MID: 0x2503
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
creating file c:\pwned.txt on the target
Done
```



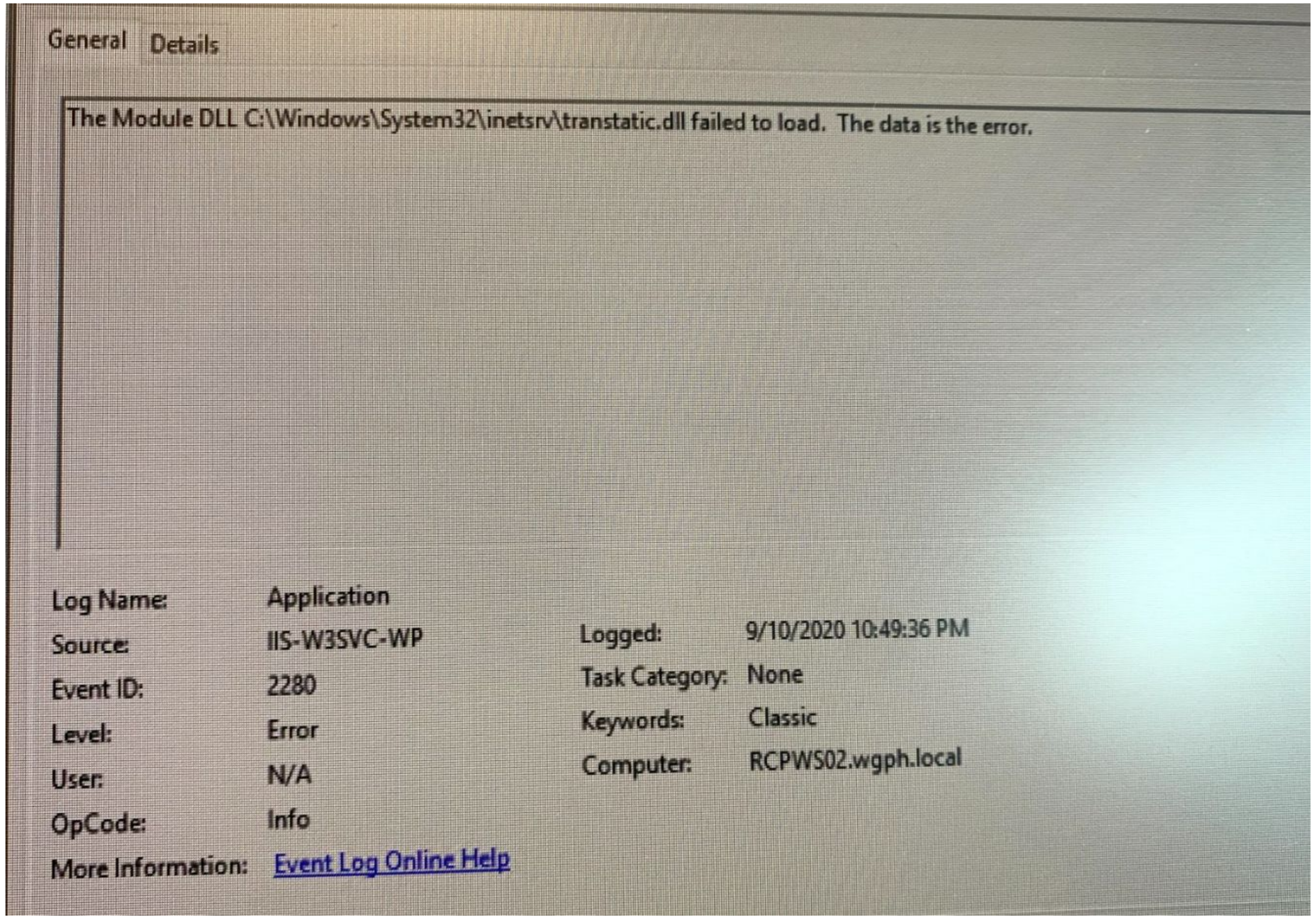
```
user:[redacted] rid:[0x94d]
user:[redacted] rid:[0x94f]
user:[redacted] rid:[0x950]
user:[alanhoho] rid:[0x954]
user:[alanho2] rid:[0x955]
user:[redacted] rid:[0xc1e]
user:[redacted] rid:[0xc1f]
```

First Conclusion

- SSL VPN exploit and it lead to internal network intrusion, and AD could possibly be owned.
- There could be laterally movement from the attackers, other servers could be compromised too.
- A hacking activity was reported a month later, and we checked the Splunk logs, and we drilled down into the suspicious servers, we found that the hacking activities were still happening.
- IIS Server-01 was the infected server.
- IIS Server-02 was the new server to replace Server-01 after the hacking activity reported.

Investigation on IIS Servers

When IIS Server-02 was newly installed, engineers found that the IIS kept throwing errors.



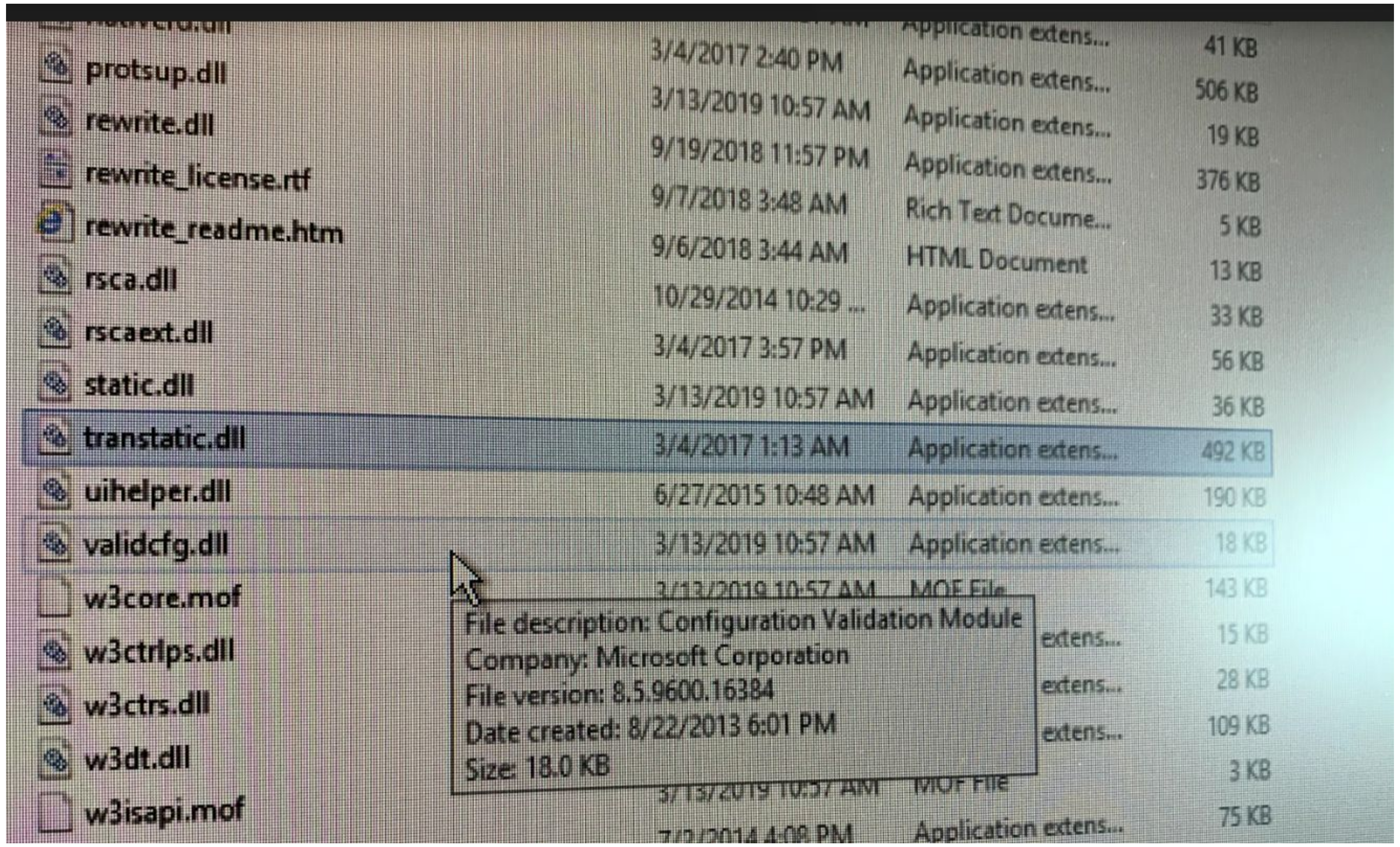
```
The Module DLL C:\Windows\System32\inetsrv\transtatic.dll failed to load.  The data is the error.

Log Name:         Application
Source:           IIS-W3SVC-WP           Logged:         9/10/2020 10:49:36 PM
Event ID:         2280                   Task Category:  None
Level:            Error                  Keywords:       Classic
User:             N/A                    Computer:       RCPWS02.wgph.local
OpCode:           Info
More Information: Event Log Online Help
```

Why IIS kept throwing error

- Engineers reported that they installed Server-02 by cloning the same configuration files from the old server.
- The IIS error implied that the dll was missing and could not be loaded in IIS.

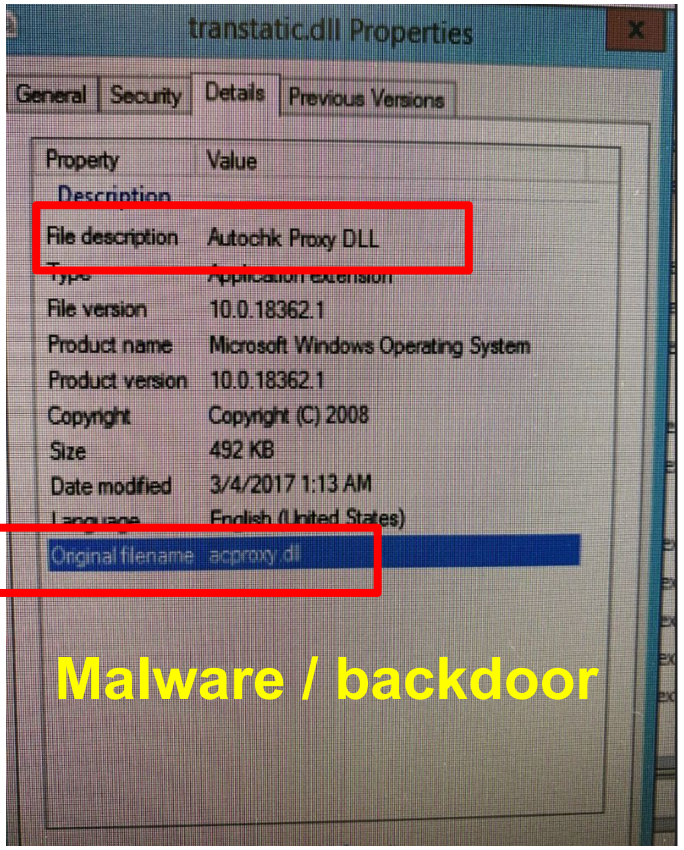
Checking IIS Server-01



		Application extens...	41 KB
	3/4/2017 2:40 PM	Application extens...	506 KB
protsup.dll	3/13/2019 10:57 AM	Application extens...	19 KB
rewrite.dll	9/19/2018 11:57 PM	Application extens...	376 KB
rewrite_license.rtf	9/7/2018 3:48 AM	Rich Text Docume...	5 KB
rewrite_readme.htm	9/6/2018 3:44 AM	HTML Document	13 KB
rsca.dll	10/29/2014 10:29 ...	Application extens...	33 KB
rscaext.dll	3/4/2017 3:57 PM	Application extens...	56 KB
static.dll	3/13/2019 10:57 AM	Application extens...	36 KB
transtatic.dll	3/4/2017 1:13 AM	Application extens...	492 KB
uihelper.dll	6/27/2015 10:48 AM	Application extens...	190 KB
validcfg.dll	3/13/2019 10:57 AM	Application extens...	18 KB
w3core.mof	3/13/2019 10:57 AM	MOF File	143 KB
w3ctrlps.dll		extens...	15 KB
w3ctrs.dll		extens...	28 KB
w3dt.dll		extens...	109 KB
w3isapi.mof	3/13/2019 10:57 AM	MOF File	3 KB
	7/2/2014 4:08 PM	Application extens...	75 KB

```
File description: Configuration Validation Module
Company: Microsoft Corporation
File version: 8.5.9600.16384
Date created: 8/22/2013 6:01 PM
Size: 18.0 KB
```

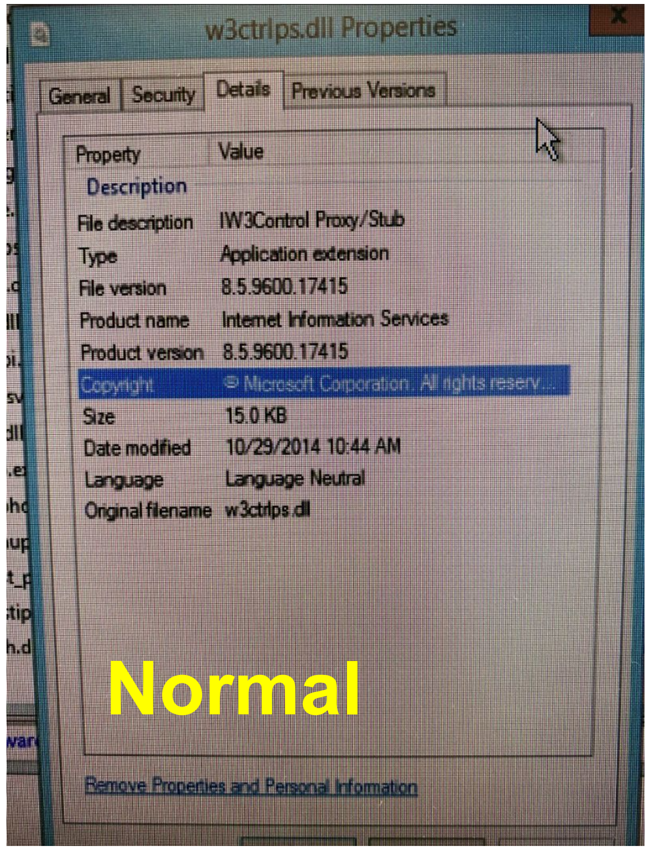
Checking IIS Server-01


transtatic.dll Properties

General | Security | Details | Previous Versions

Property	Value
Description	
File description	Autochk Proxy DLL
Type	Application extension
File version	10.0.18362.1
Product name	Microsoft Windows Operating System
Product version	10.0.18362.1
Copyright	Copyright (C) 2008
Size	492 KB
Date modified	3/4/2017 1:13 AM
Language	English (United States)
Original filename	acproxy.dll

Malware / backdoor


w3ctrlps.dll Properties

General | Security | Details | Previous Versions

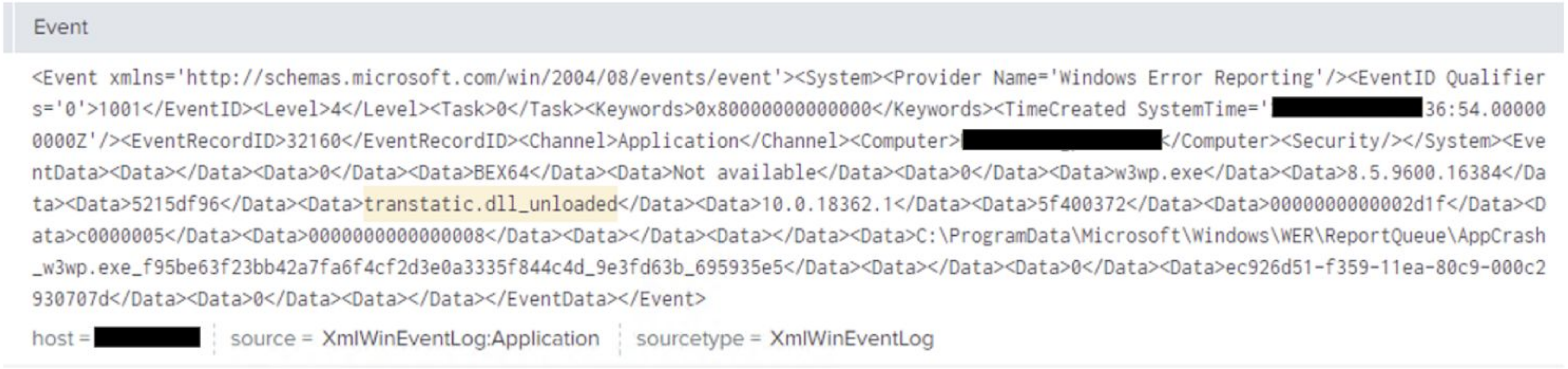
Property	Value
Description	
File description	IW3Control Proxy/Stub
Type	Application extension
File version	8.5.9600.17415
Product name	Internet Information Services
Product version	8.5.9600.17415
Copyright	© Microsoft Corporation. All rights reserv...
Size	15.0 KB
Date modified	10/29/2014 10:44 AM
Language	Language Neutral
Original filename	w3ctrlps.dll

Normal

Remove Properties and Personal Information

Tracing the error

It is found that dll was attempted to be unloaded.



Event

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Windows Error Reporting'/><EventID Qualifier
s='0'>1001</EventID><Level>4</Level><Task>0</Task><Keywords>0x80000000000000</Keywords><TimeCreated SystemTime='[redacted]36:54.00000
0000Z'/><EventRecordID>32160</EventRecordID><Channel>Application</Channel><Computer>[redacted]</Computer><Security/></System><Eve
ntData><Data></Data><Data>0</Data><Data>BEX64</Data><Data>Not available</Data><Data>0</Data><Data>w3wp.exe</Data><Data>8.5.9600.16384</Da
ta><Data>5215df96</Data><Data>transtatic.dll_unloaded</Data><Data>10.0.18362.1</Data><Data>5f400372</Data><Data>000000000002d1f</Data><D
ata>c0000005</Data><Data>0000000000000008</Data><Data></Data><Data></Data><Data>C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash
_w3wp.exe_f95be63f23bb42a7fa6f4cf2d3e0a3335f844c4d_9e3fd63b_695935e5</Data><Data></Data><Data>0</Data><Data>ec926d51-f359-11ea-80c9-000c2
930707d</Data><Data>0</Data><Data></Data></EventData></Event>
```

host = [redacted] | source = XmlWinEventLog:Application | sourcetype = XmlWinEventLog

Tracing the error

- We could not find the malware (transtatic.dll) in Server-02, only found in Server-01.
- The IIS error in Server-02, because the IIS configuration (applicationHost.config) tried to load the malware dll file, but it did not exist.

Case Summary

Attack Path:

- SSL VPN Exploitation - retrieved credentials before launching the real attack
- Active Directory Exploitation - admin accounts were created

We assumed the attackers accessed network engineers' desktop by:

- Same passwords of AD and SSL VPN
- Created temporarily domain account to login
- Firewall Admin UI Password was saved in the browser
- Laterally movement and planted IIS backdoor

Controls were applied to the servers after the incident

- enabled 2FA, applied patches (Firewall, Servers)

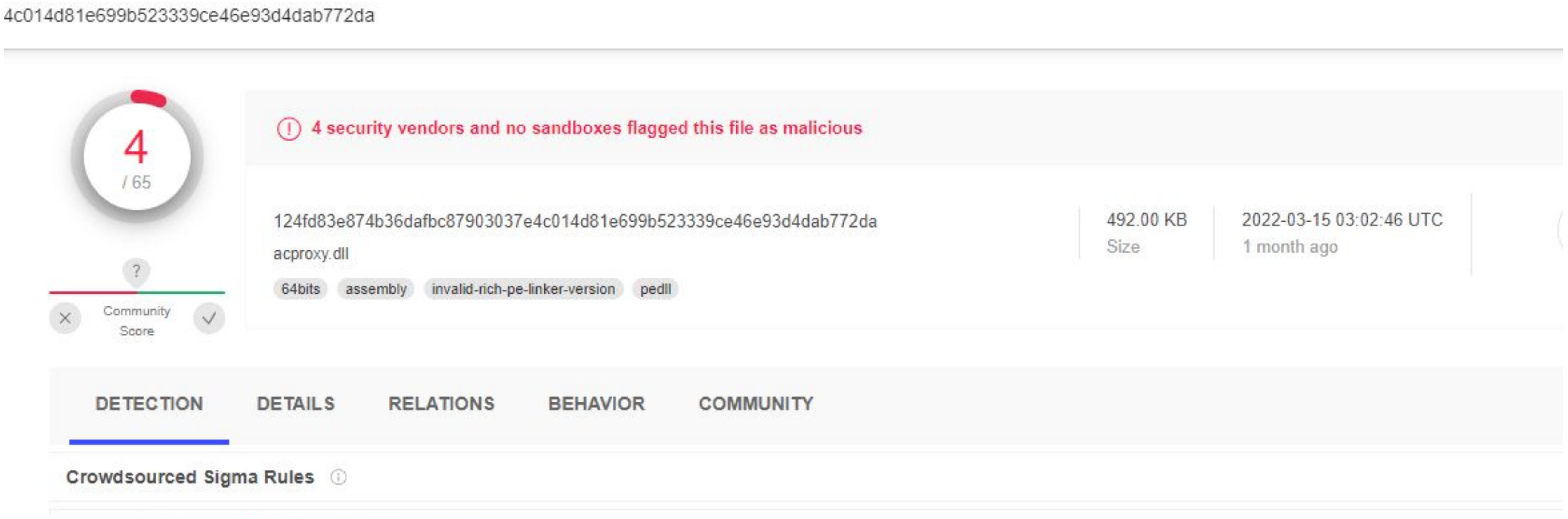
Further Analysis on IIS Backdoor

We extracted the malware (transtatic.dll) from Server-01 to see if we can discover more information about the attackers.

However, in this presentation, we focus on our defense and detection methodologies.

VirusTotal Detection

So far there are 4 detectors can identify the backdoor (22 April 2022).
In Y2020 and Y2021, there are no detection since our submission.



4c014d81e699b523339ce46e93d4dab772da

4 / 65

4 security vendors and no sandboxes flagged this file as malicious

124fd83e874b36dafbc87903037e4c014d81e699b523339ce46e93d4dab772da
acproxy.dll

64bits assembly invalid-rich-pe-linker-version pedll

492.00 KB
Size

2022-03-15 03:02:46 UTC
1 month ago

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Crowdsourced Sigma Rules

Motivation

- Typical anti-virus cannot identify highly customized and targeted attack backdoor

- We establish methodology to detect the backdoor threat and playbook for incident response.

Analysis methodology

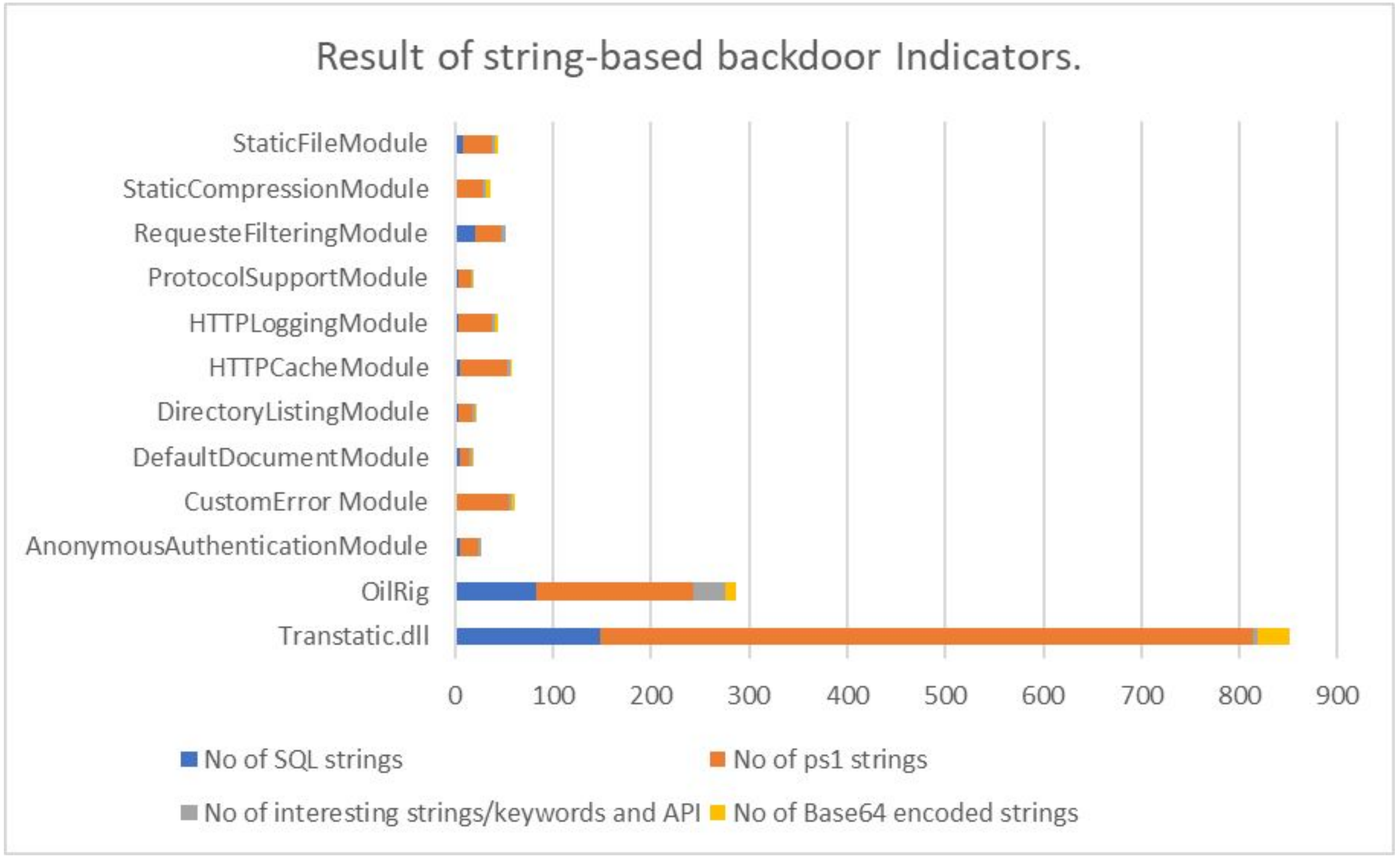
- Static analysis
  - Similarity/signature base
    - Locality sensitive hashing
    - Yara rules
    - Strings and Frequency-based
  - Entropy for obfuscation detection
  - AI based
- Dynamic analysis
  - Running in the sandbox, e.g., cuckoo sandbox
    - Sensitive API

Analysis methodology

- Static analysis
  - Similarity/signature base
    - Locality sensitive hashing
    - Yara rules
    - Strings and Frequency-based
  - Entropy for obfuscation detection
  - AI based
- Dynamic analysis
  - Running in the sandbox, e.g., cuckoo sandbox
    - Sensitive API

String and Frequency-based backdoor indicators

- We extracted strings with length large than 3 from the samples
- Performed following analysis
  - Obfuscation analysis
  - Counting on special API
  - Number of SQL and PowerShell like strings
- We compared against
  - IIS native modules deployed in production
  - RGDoor IIS backdoor samples
- A lot of PowerShell like strings presence in Transtatic sample, but not other samples

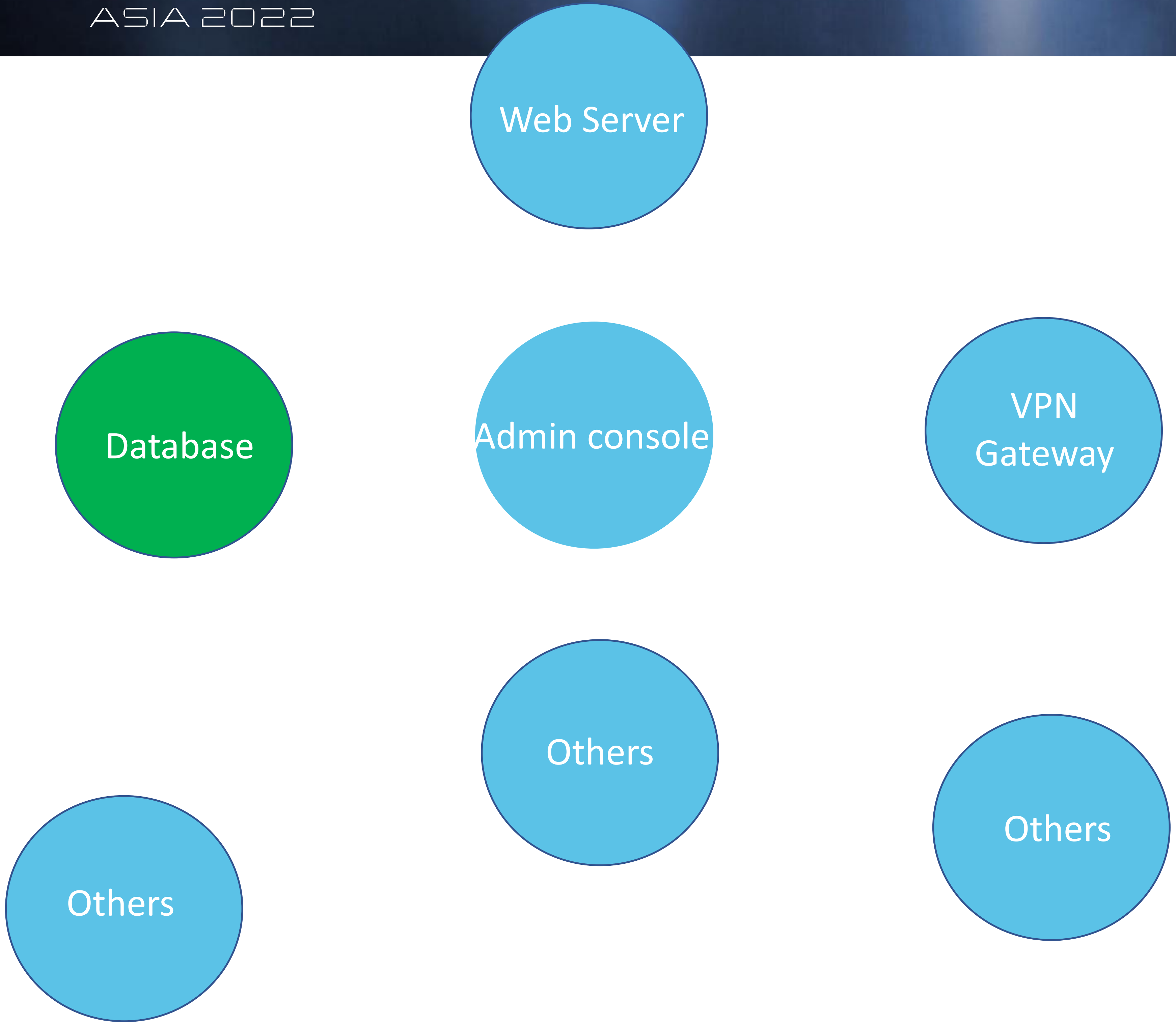


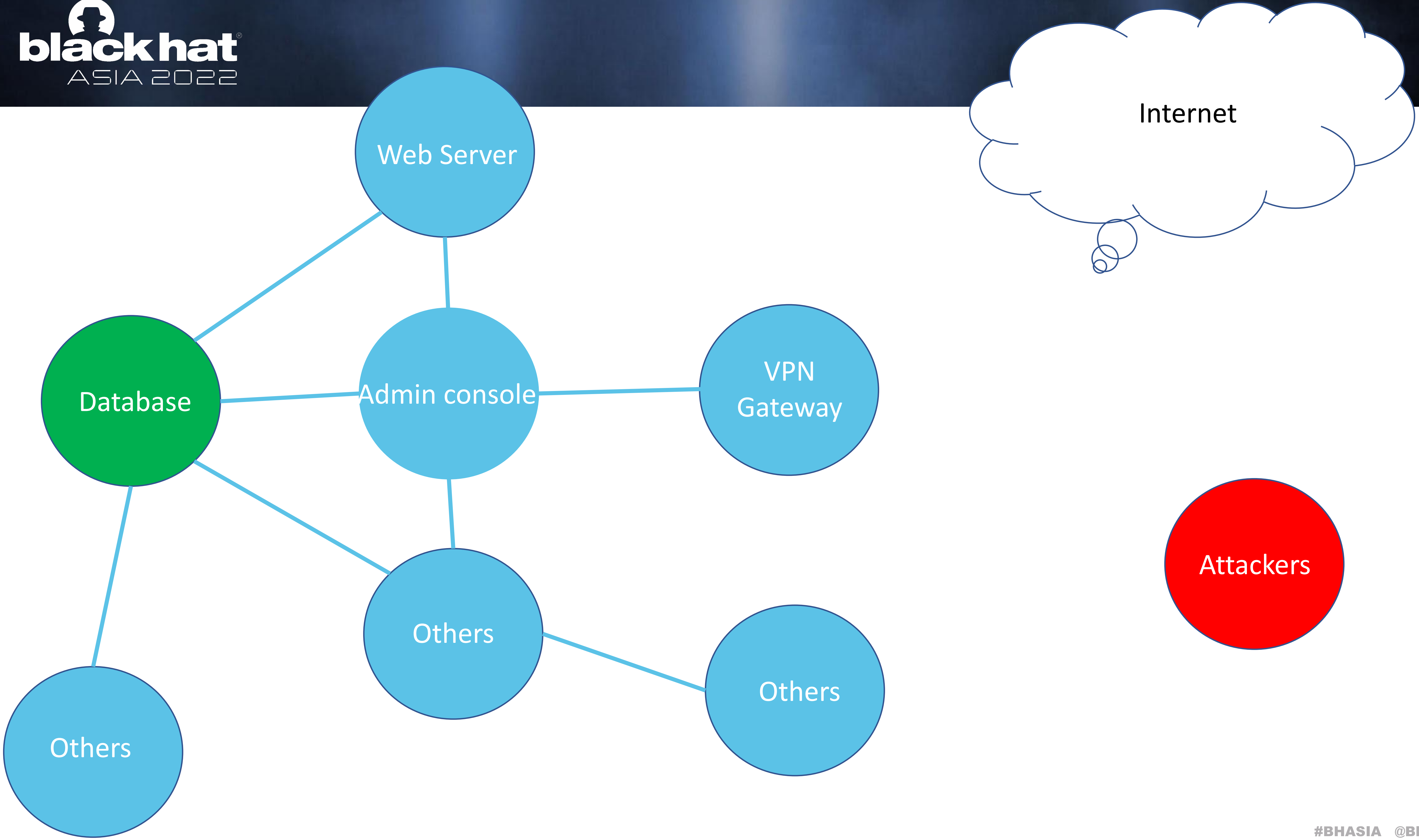
AI model

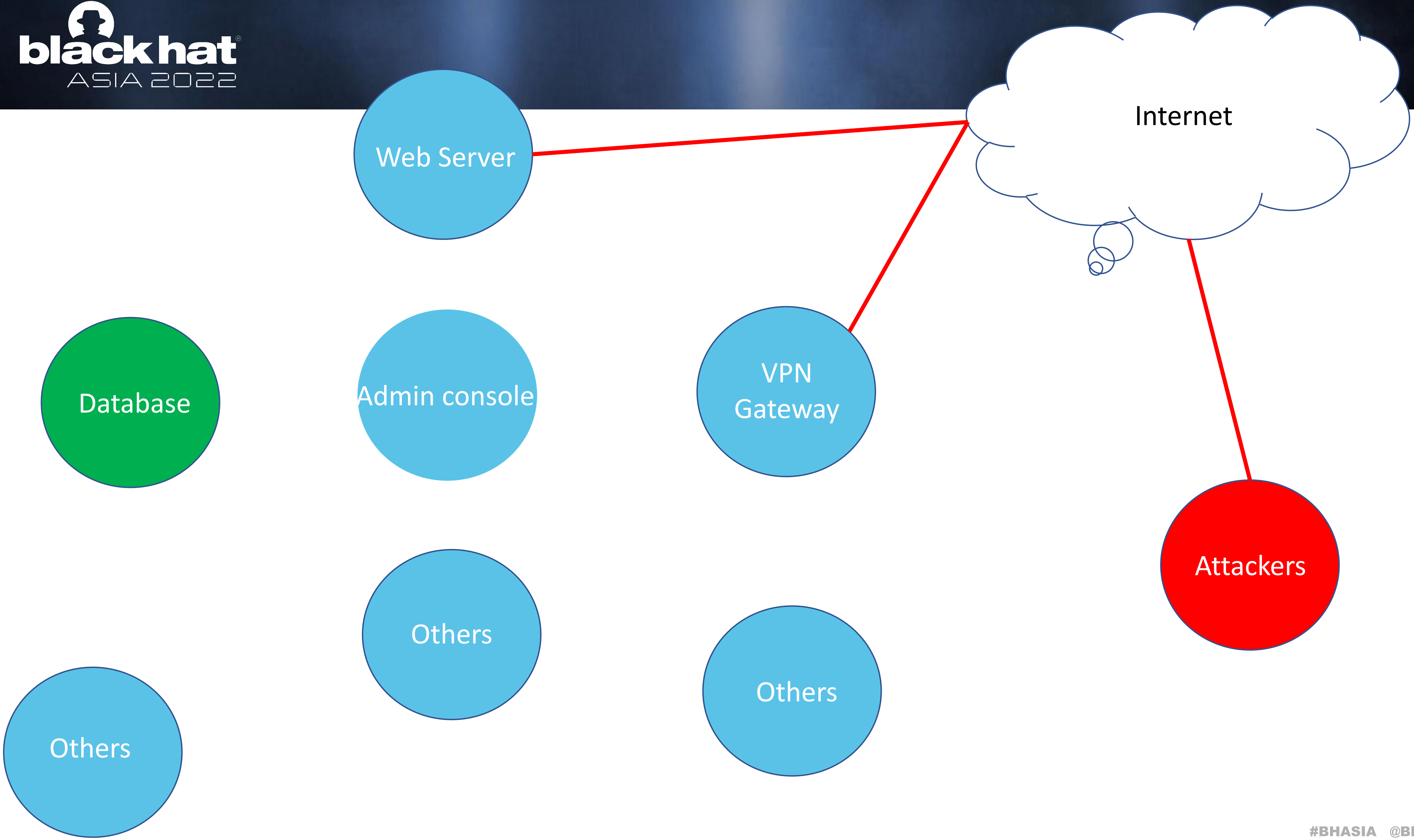
- We examined the collected samples with our in-house malware detection models.
- It works well for detecting APT samples.
- It only detect 1 out of 5 of the samples we found on the server.
- We interpreted the results as following:
    - The binary were compose with a lot of legitimate open source libraries
    - No obfuscation (for most of the component)
- Stealthiness of these set of samples against real world AV.

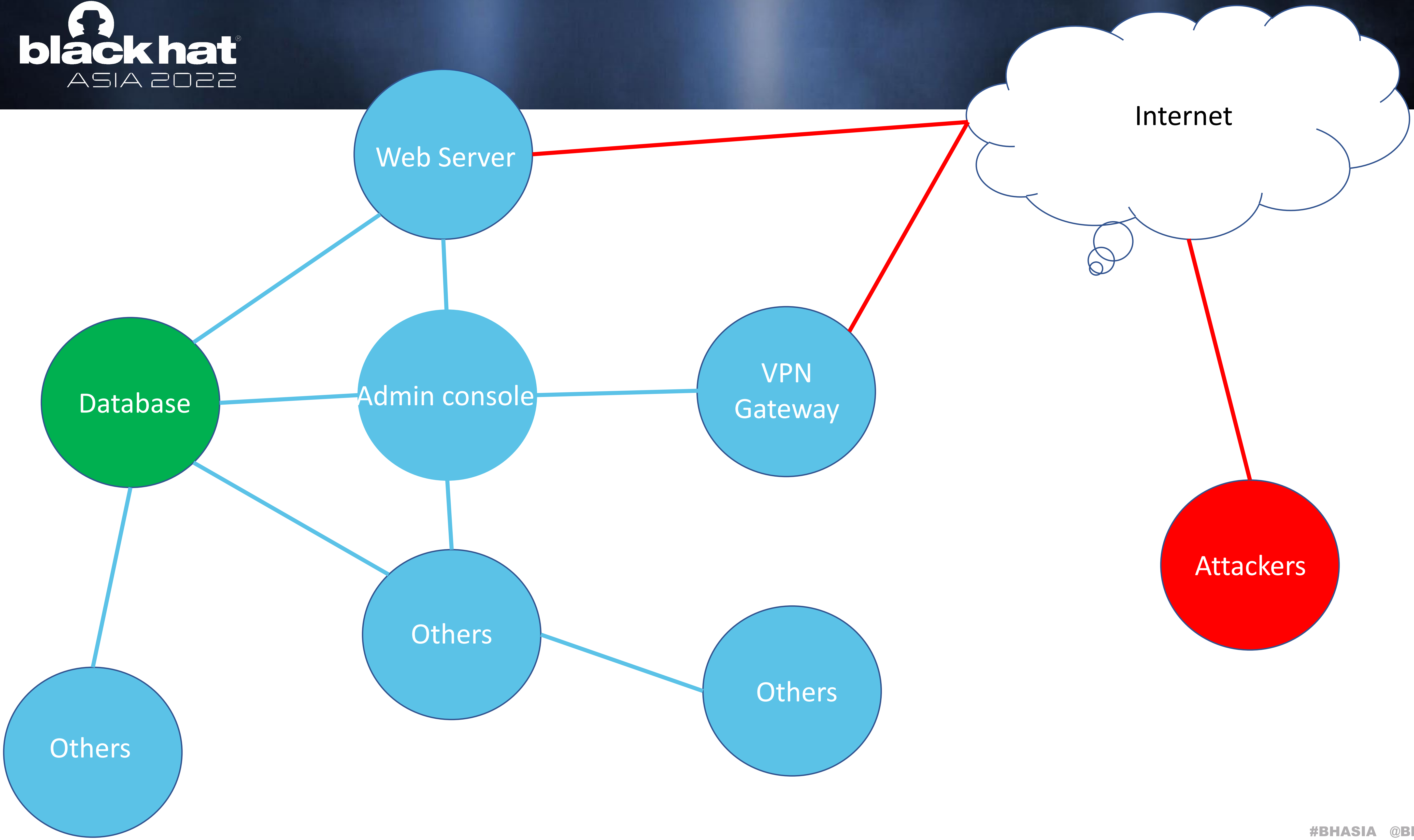
BackDoor Incident Response Algorithm

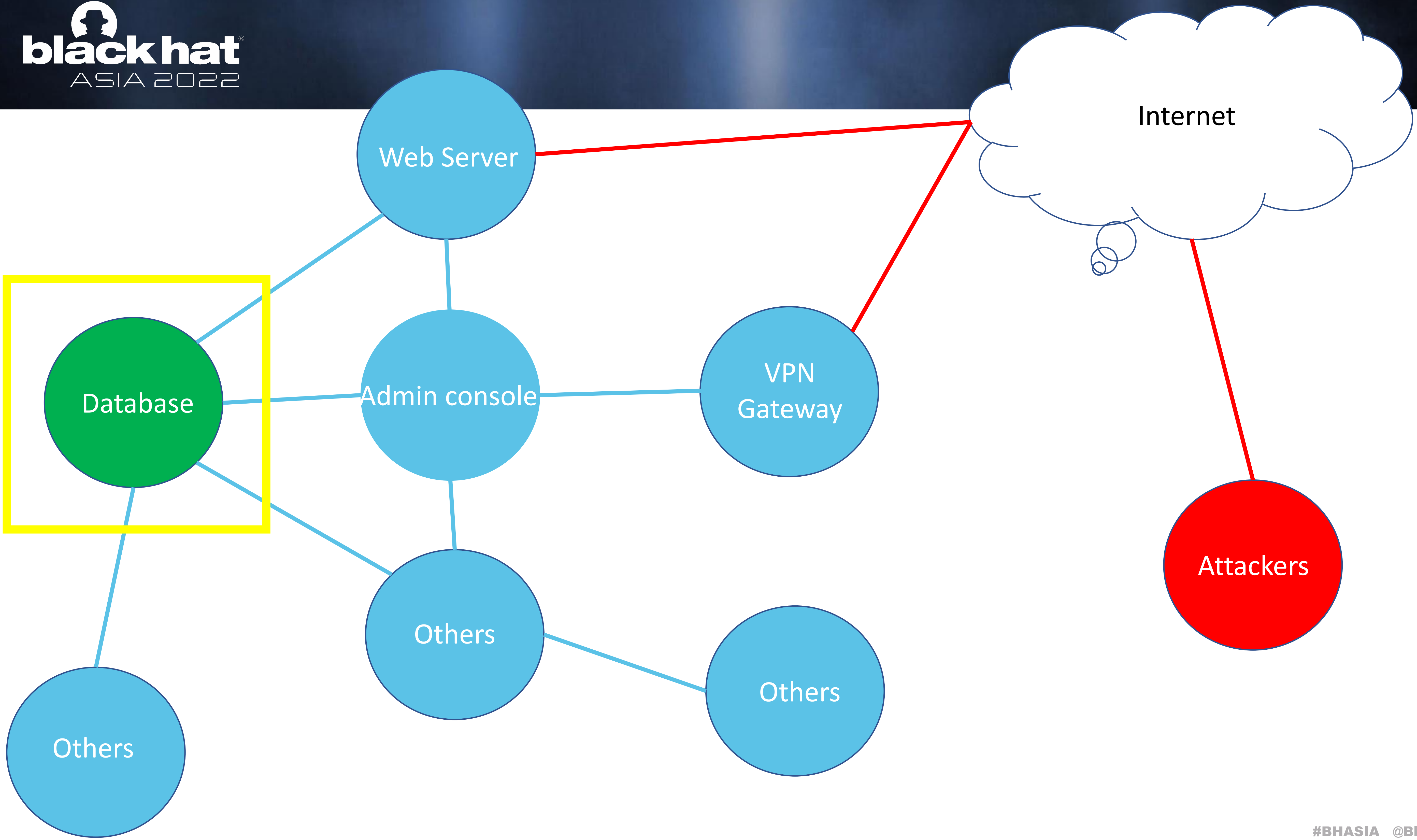
- We treat the network architecture as a graph.
- We use vertex to represent servers, workstations, etc and edge to represent connections (where blue for trusted connection, red for untrusted connection).
- We will pick a target and start the investigation.
- We check for the following criteria:
    1. Examine user authentication and suspicious operation, like Delete-Create-Execute-Delete-Create operations.
    2. Ratio of accounts used to authenticate to different systems to the number of authenticated systems connected.
    3. Configuration difference from the original.
- Further investigation if meet any of the above conditions.

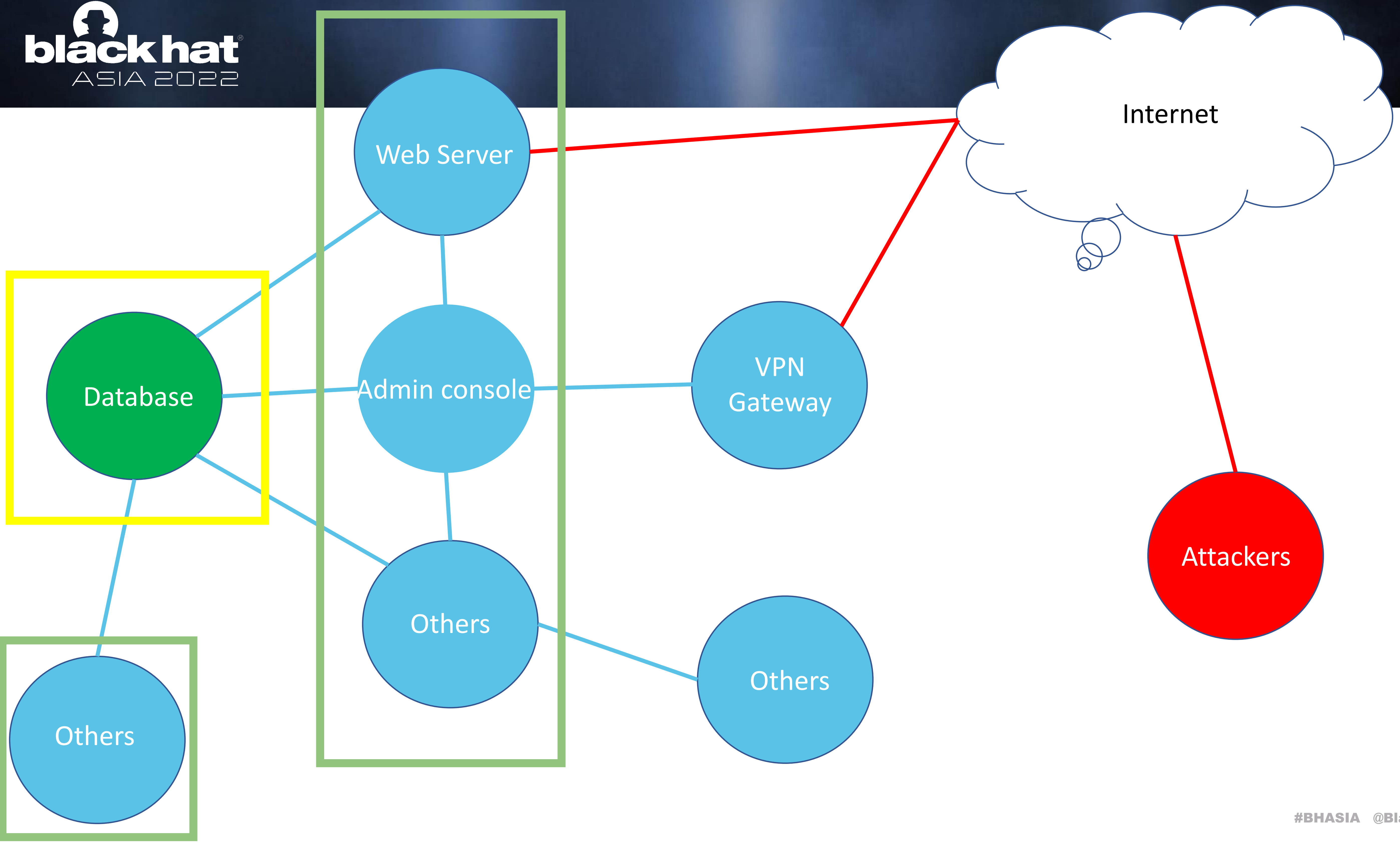

Web Server
Database
Admin console
VPN Gateway
Others
Others
Others

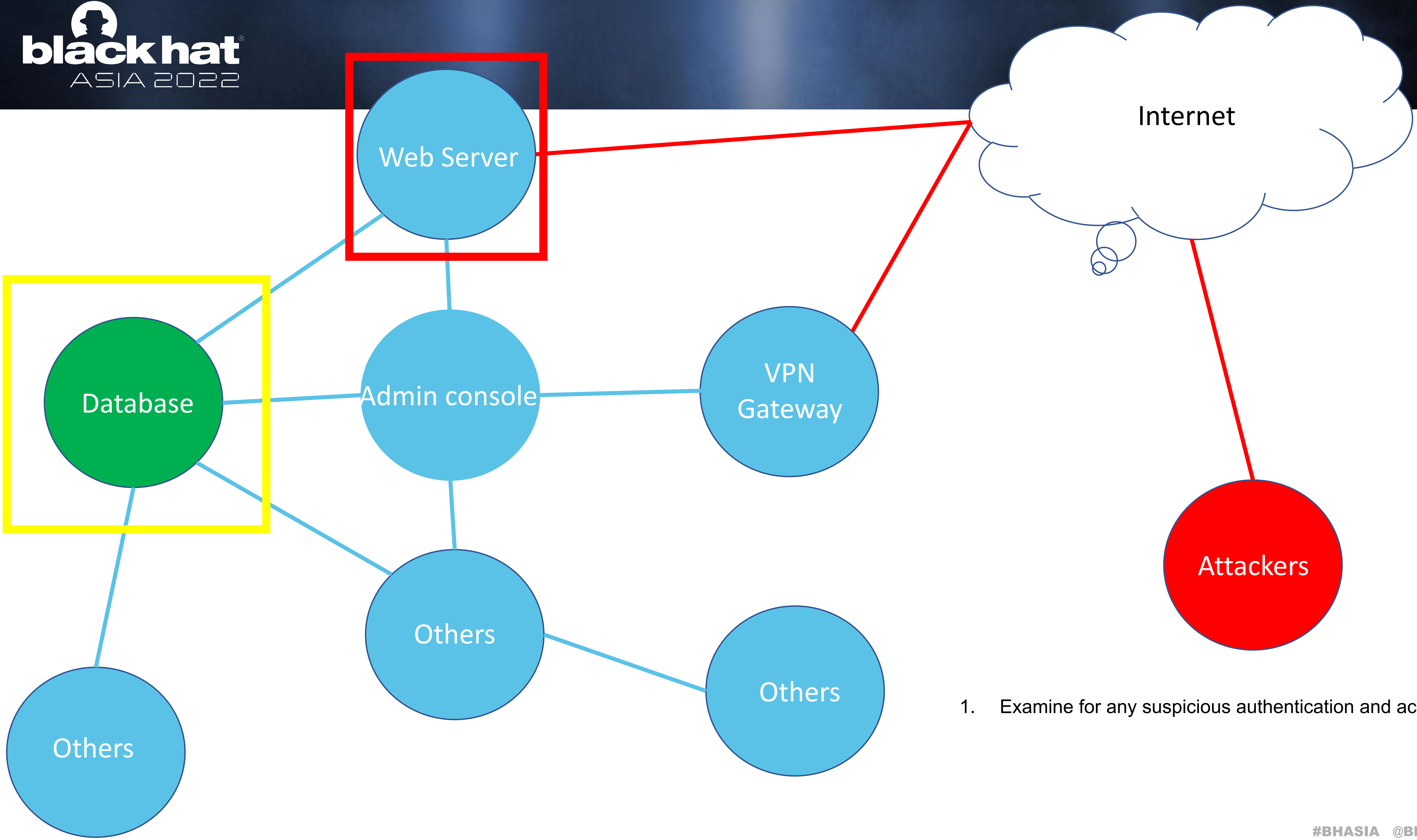

Internet
Web Server
Database
Admin console
VPN Gateway
Others
Others
Others
Attackers


Internet
Web Server
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others

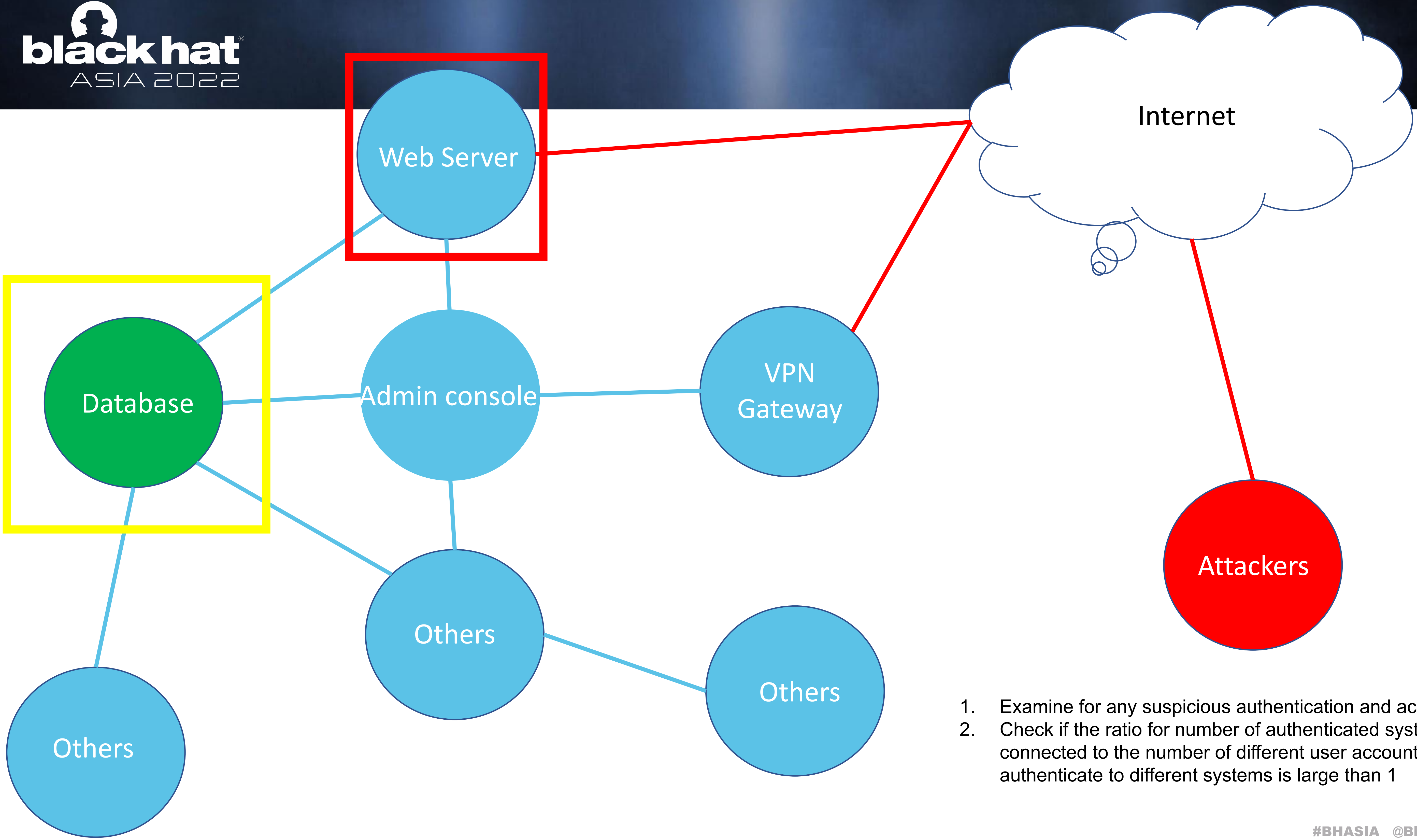

Internet
Web Server
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others


Internet
Web Server
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others

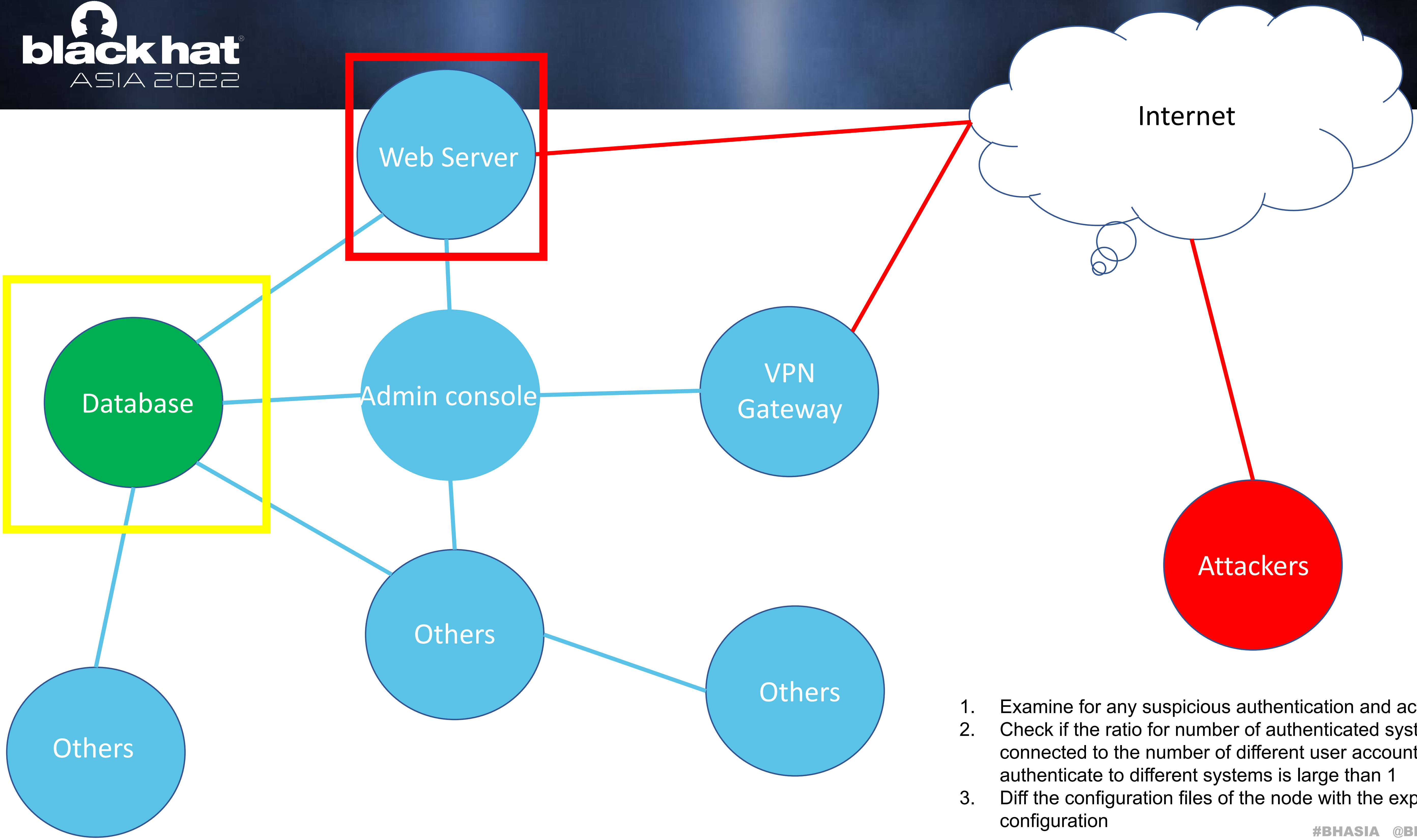

Web Server
Internet
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others



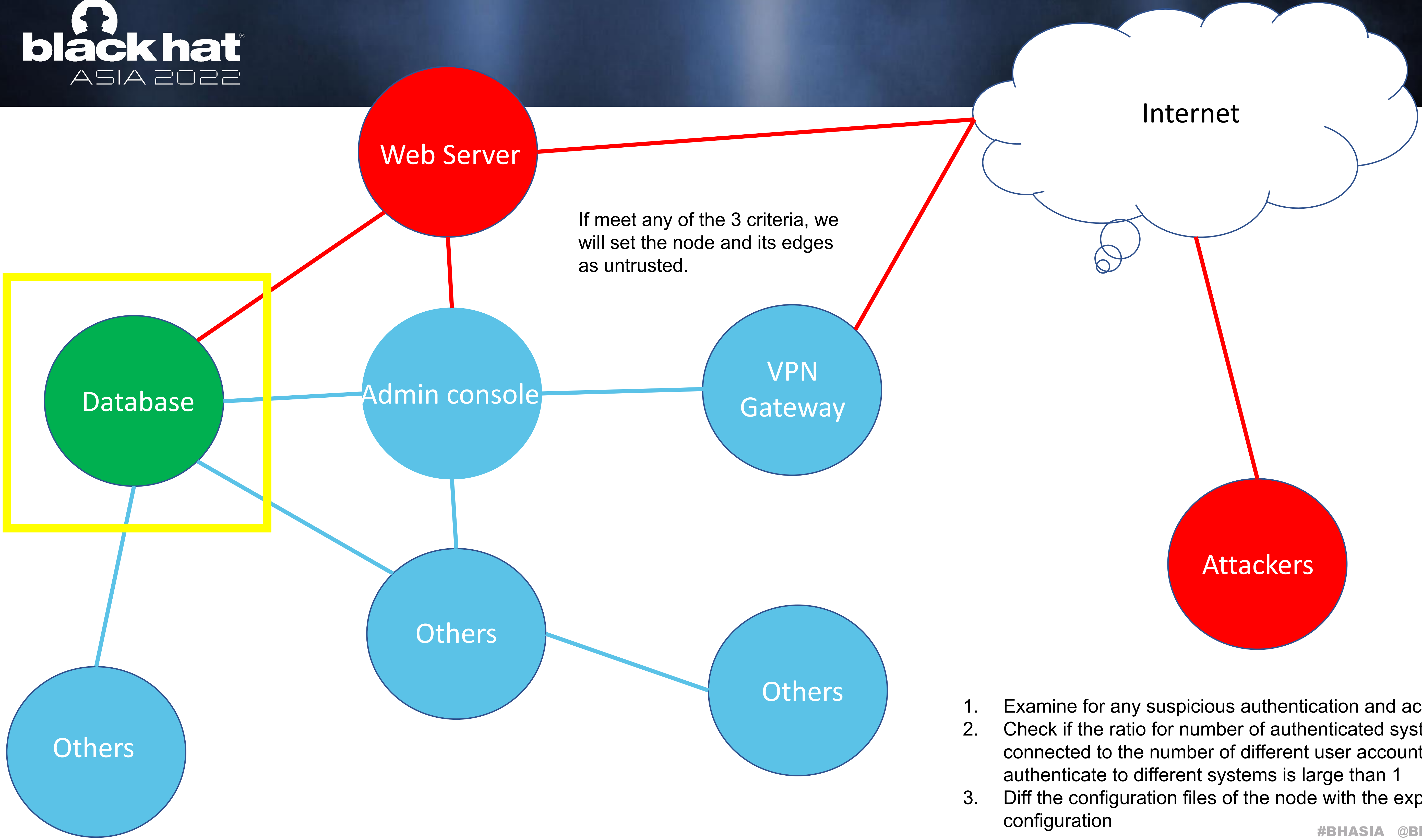
1. Examine for any suspicious authentication and activities



1. Examine for any suspicious authentication and activities
2. Check if the ratio for number of authenticated systems connected to the number of different user accounts authenticate to different systems is large than 1

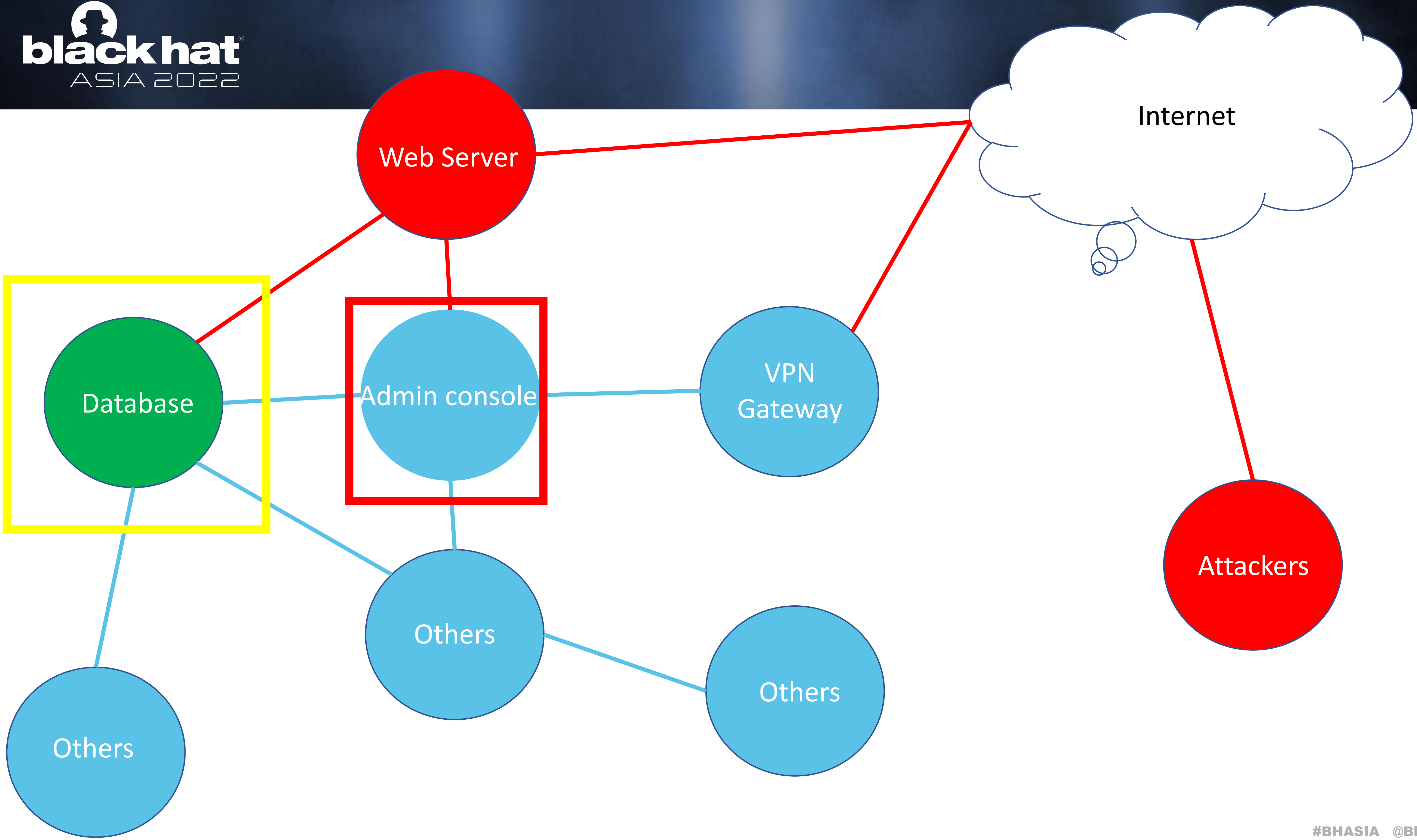


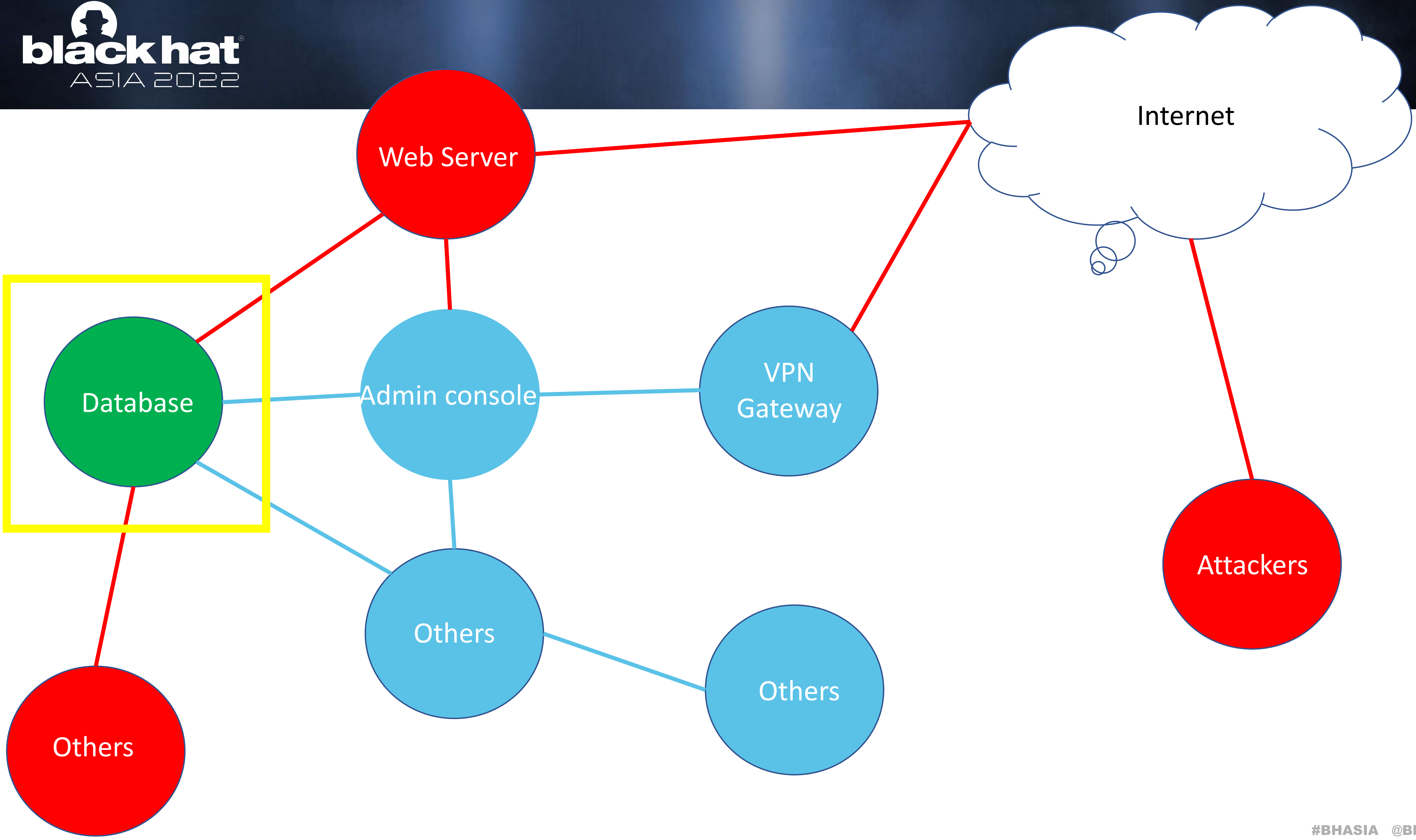
1. Examine for any suspicious authentication and activities
2. Check if the ratio for number of authenticated systems connected to the number of different user accounts authenticate to different systems is large than 1
3. Diff the configuration files of the node with the expected configuration

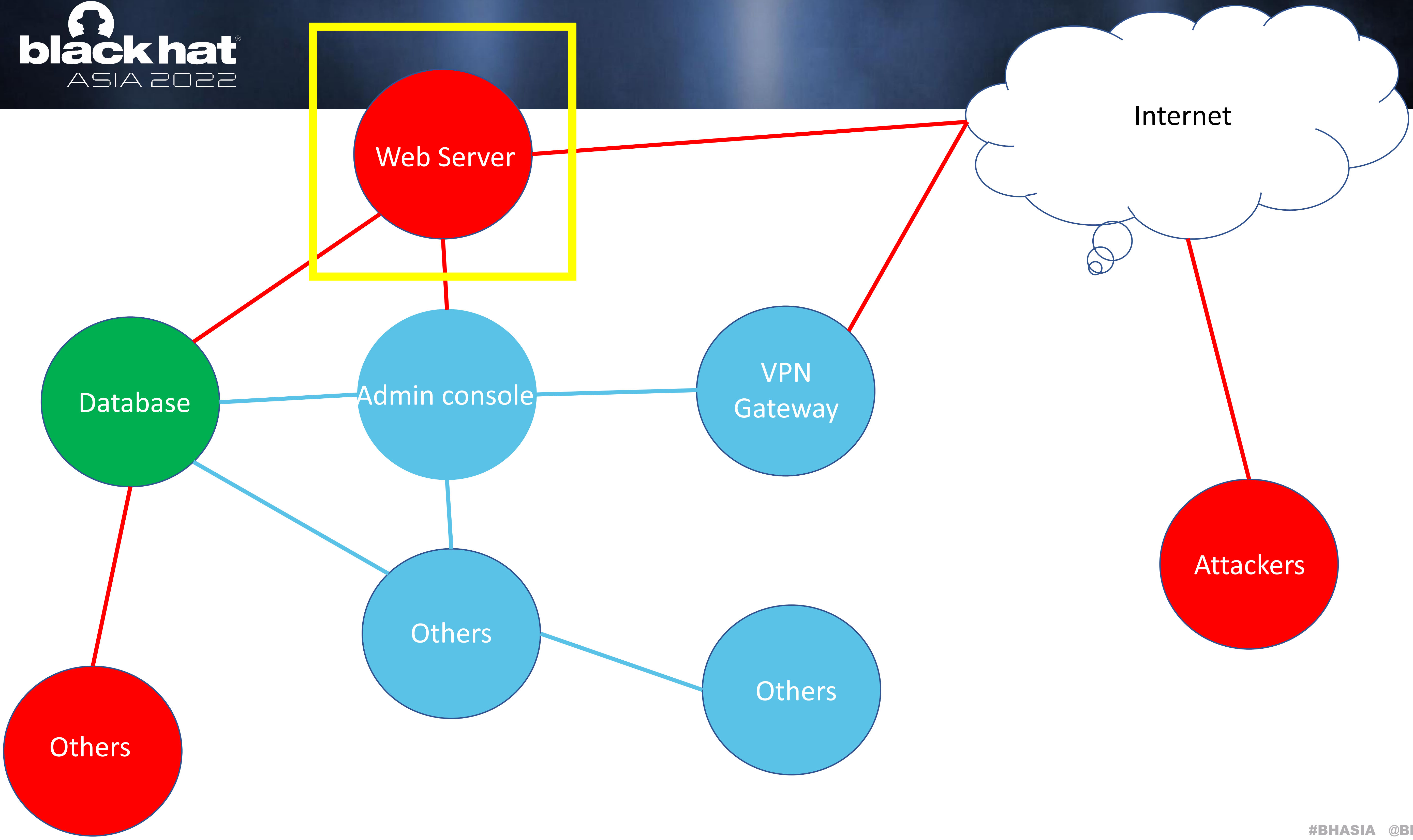


If meet any of the 3 criteria, we will set the node and its edges as untrusted.

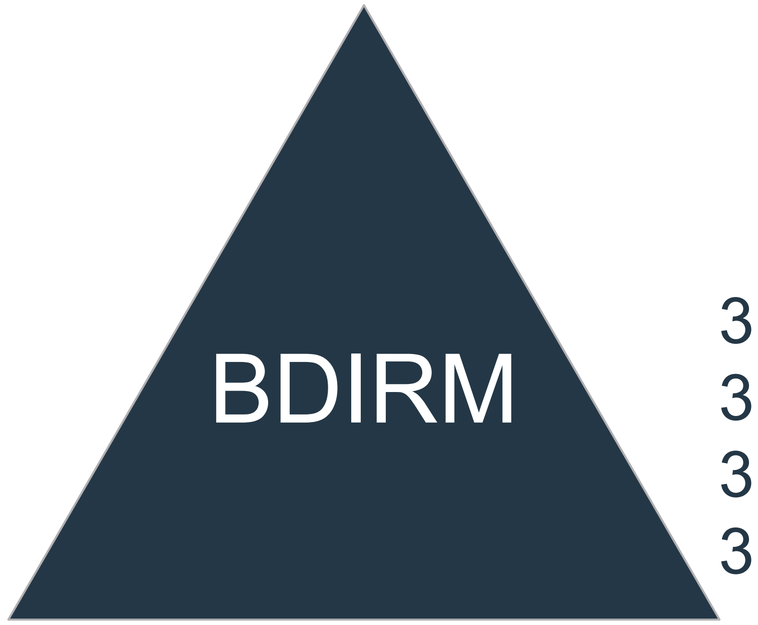
1. Examine for any suspicious authentication and activities
2. Check if the ratio for number of authenticated systems connected to the number of different user accounts authenticate to different systems is large than 1
3. Diff the configuration files of the node with the expected configuration


Web Server
Internet
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others


Web Server
Internet
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others


Web Server
Internet
Database
Admin console
VPN Gateway
Attackers
Others
Others
Others

Backdoor Incident Response Matrix



Backdoored Server

2.1. Master Config Preparation
2.2. Master Config Comparison
2.3. Activity Log Analysis
2.4. Memory Acquisition

Network Traffic

3.1. Network Traffic Benchmarking
3.2. Network Traffic Anomaly Detection
3.3. Network Traffic Search Rule
3.4. Network Traffic Deobfuscation

Backdoor

1.1. Functionality Analysis
1.2. Similarity Analysis
1.3. String and Frequency Analysis
1.4. Entropy Analysis

BDIRM - Backdoor

1.1 Functionality Analysis

a. Identify the function capability of the suspicious backdoor, which is helpful to reverse engineering.
b. Carry out static analysis and reverse engineering of the backdoor binary or/and webshell.
   - ➢ It helps to update the network traffic search rules in 3.3.
c. Emulate the binary in emulation framework for dynamic analysis.
d. Write Yara rules with artifacts found from items a – c. It will be useful to scan and discover any backdoors in other nodes of machines or systems.

BDIRM - Backdoor (Contd.)

1.2 Similarity Analysis

- Carry out similarity checks with published backdoor whether they share similar code instruction structure and commands with binary differing and TLSH.
  - ➢ It helps to update the network traffic search rules in 3.3.

1.3 String and Frequency Analysis

- Scan all DLLs to detect any abnormally high number of encoded base64 strings, obfuscated SQL and Powershell scripts and possible system command(s) and file upload/download/delete command(s).

1.4 Entropy Analysis

- Apply Shannon entropy detection over the binary to detect obfuscated, compressed and/or encrypted code distribution.

BDIRM - Backdoored Server

2.1 Prepare the golden copy of configuration and deployment files.

2.2 Compare the golden copy of configuration and deployment files with those with the potentially hacked server.

2.3 Examine activity and error logs of the hacked server to detect any successful or failed loading of suspicious external programs or/and modules.

2.4 Export and dump the volatile memory of the backdoored server.

Repeat 1.3, scan and detect any abnormally high number of encoded base64 strings, obfuscated SQL and Powershell scripts and possible system command(s) and file upload/download/delete command(s).

BDIRM - Network Traffic

3.1 Benchmark the common request/response header.

- ➢ Prerequisite: SIEM or proper logging is deployed.

3.2 Monitor high volume of HTTP request/response, and unpopular incoming and outgoing IP addresses.

- ➢ Prerequisite: SIEM or proper logging is deployed

3.3 Network Traffic Search Rule

a) Write search rules to detect and alert unpopular user agent and cookie values from the logs.
b) Write search rules to detect and alert suspicious requests based on findings in 1.1, 1.2, 1.3 and 2.4.

3.4 Network traffic deobfuscation: Detect encrypted and encoded traffic.

Conclusion

- With this handled realistic case, we propose a playbook to respond to Backdoor Incident, which is a trigger-based threat with least footprints we can capture.
- Typical IR procedure is still useful but not for backdoor.
- All tools are published and known but there are not many existing systematic solutions for backdoor detection.
- How about adopting the following alternative measures?
  - ➢ Whitelisting control and hardening of OS.
  - ➢ Checking the native module configuration files.
- May not be feasible in the case of insider attack and the machine is completely controlled and managed by the attacker.