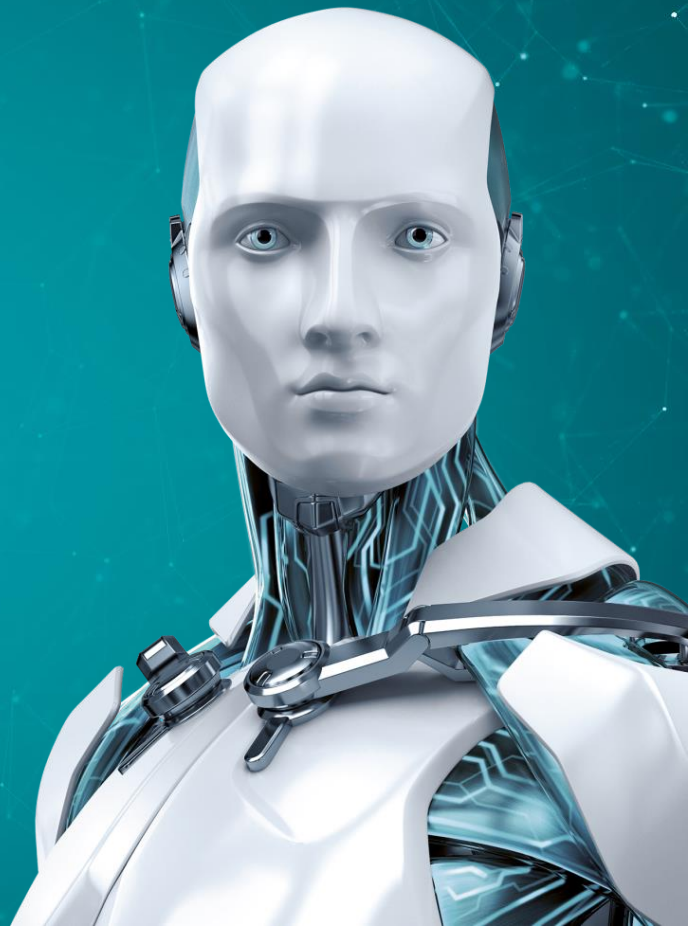
ENJOY SAFER TECHNOLOGY™

Anatomy
of native IIS malware



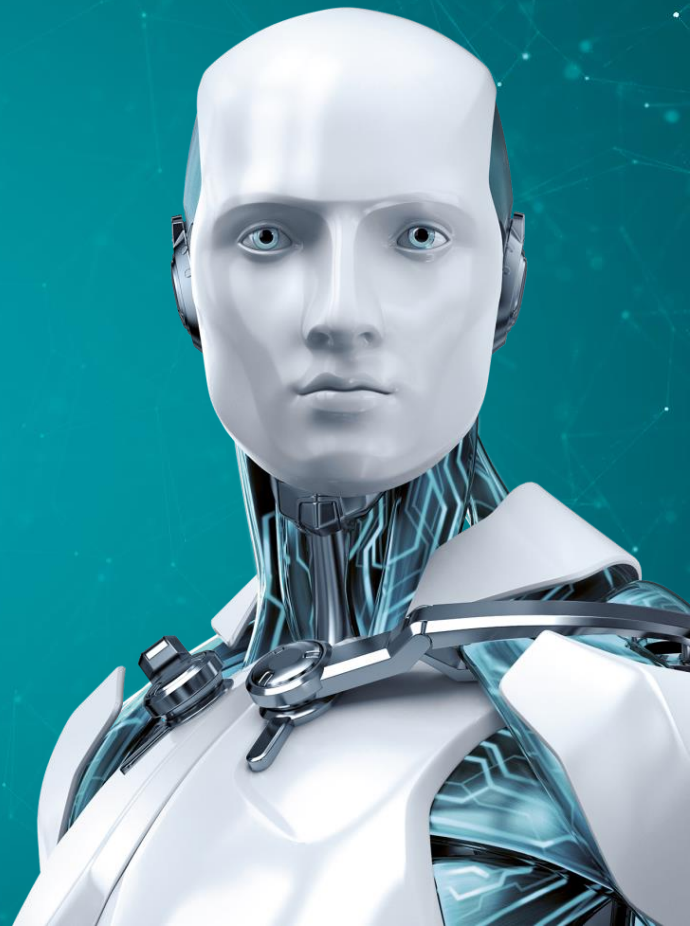
Zuzana Hromcova | Malware Researcher

ENJOY SAFER TECHNOLOGY™

Anatomy
of native IIS malware

C++ libraries

Internet Information Services



Agenda

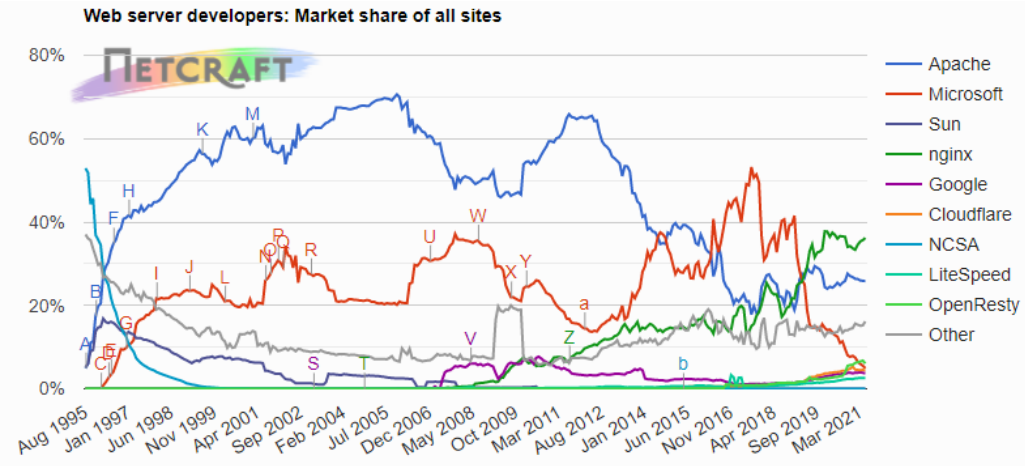
- IIS (and IIS malware) architecture
- Reversing native IIS malware
- IIS malware landscape
- Useful resources

Internet Information Services (IIS)

4-7%

of websites use
IIS software*

*Netcraft: May 2021 Web Server Survey



Developer	April 2021	Percent	May 2021	Percent
nginx	432,167,302	35.65%	440,997,336	36.19%
Apache	313,948,741	25.90%	314,774,492	25.83%
OpenResty	81,935,391	6.76%	73,839,970	6.06%
Microsoft	67,182,740	5.54%	60,265,118	4.95%

4-7%

of websites use IIS software*

*W3Techs: Usage statistics of web servers

Technologies > Web Servers

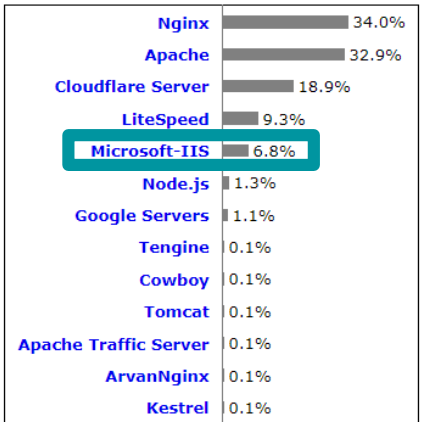
Usage statistics of web servers

This diagram shows the percentages of websites using various web servers. See technologies overview for explanations on the methodologies used in the surveys. Our reports are updated daily.

Request an extensive web servers market report.

Learn more

How to read the diagram:
Nginx is used by 34% of all the websites whose web server we know.



4-7%

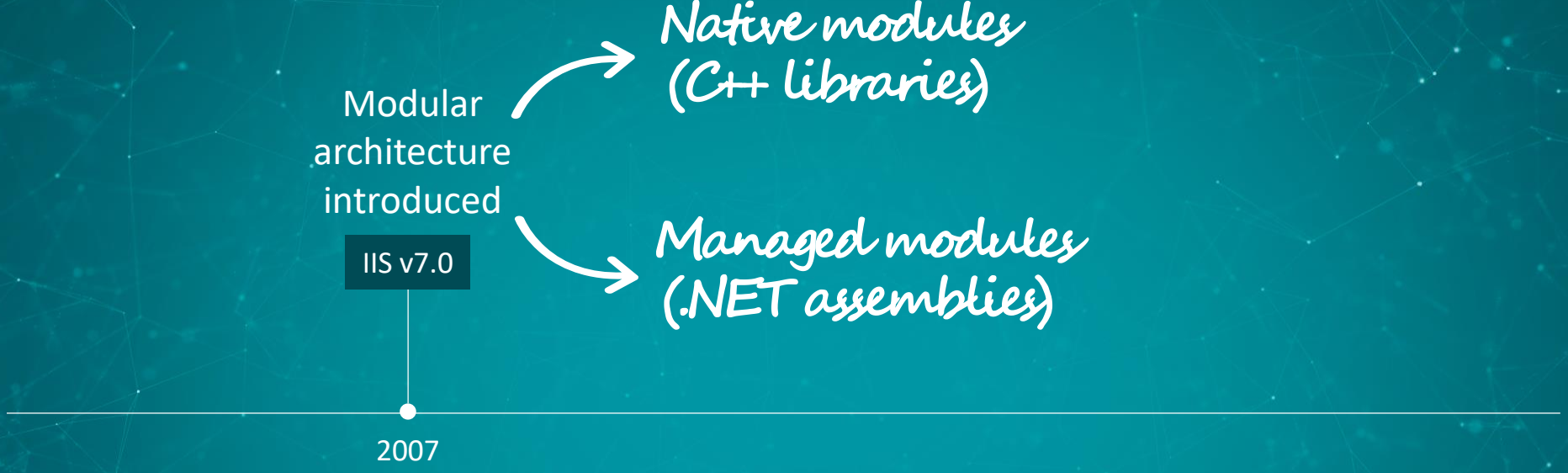
of websites use IIS software*

Microsoft Exchange
email servers with

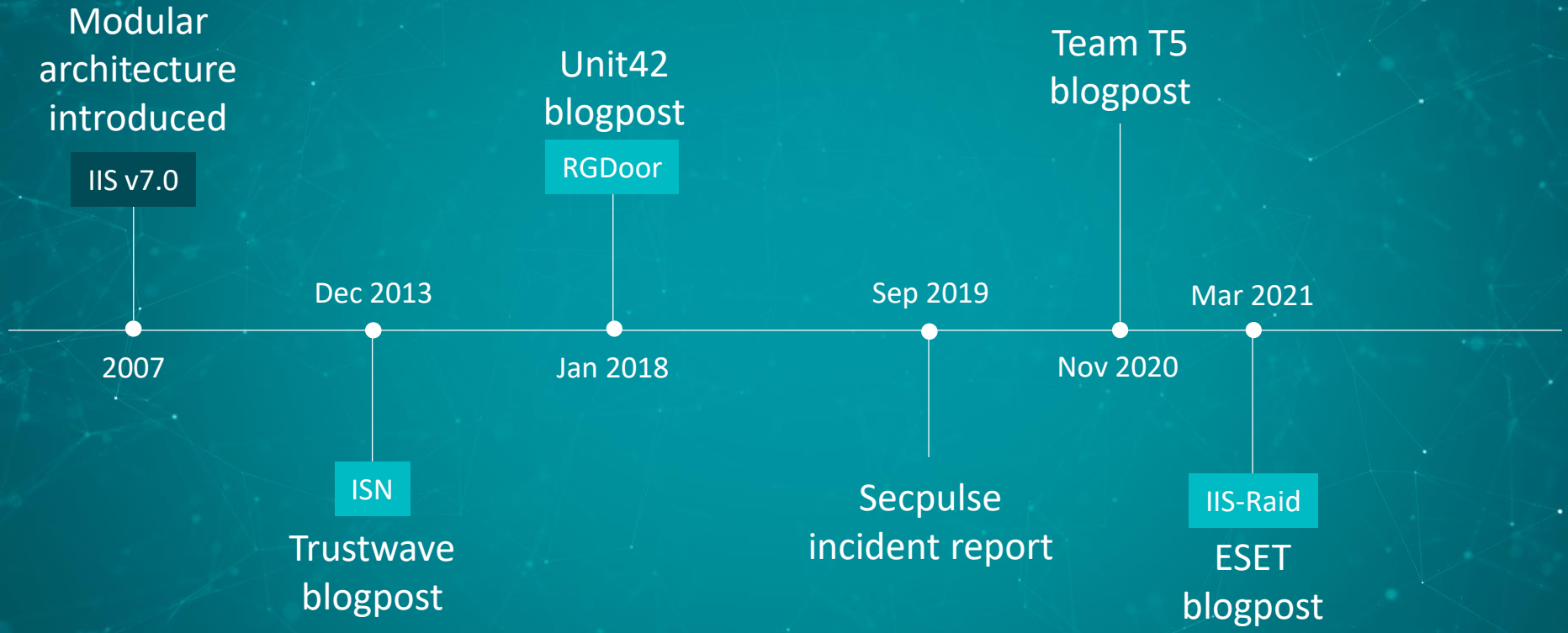
OWA

Microsoft Exchange email servers with OWA

Shodan result for public servers with OWA running Microsoft Exchange 2013 or 2016 (query for the IIS banner `X-AspNet-Version` and `Outlook` in the title):


Native modules
(C++ libraries)
Modular
architecture
introduced
IIS v7.0
Managed modules
(.NET assemblies)
2007

IIS malware reports



Exchange servers under siege from at least 10 APT groups

ESET Research has found LuckyMouse, Tick, Winnti Group, and Calypso, among others, are likely using the recent Microsoft Exchange vulnerabilities to compromise email servers all around the world

 Matthieu Faou
 Mathieu Tartare
 Thomas Dupuy

10 Mar 2021 - 02:00PM

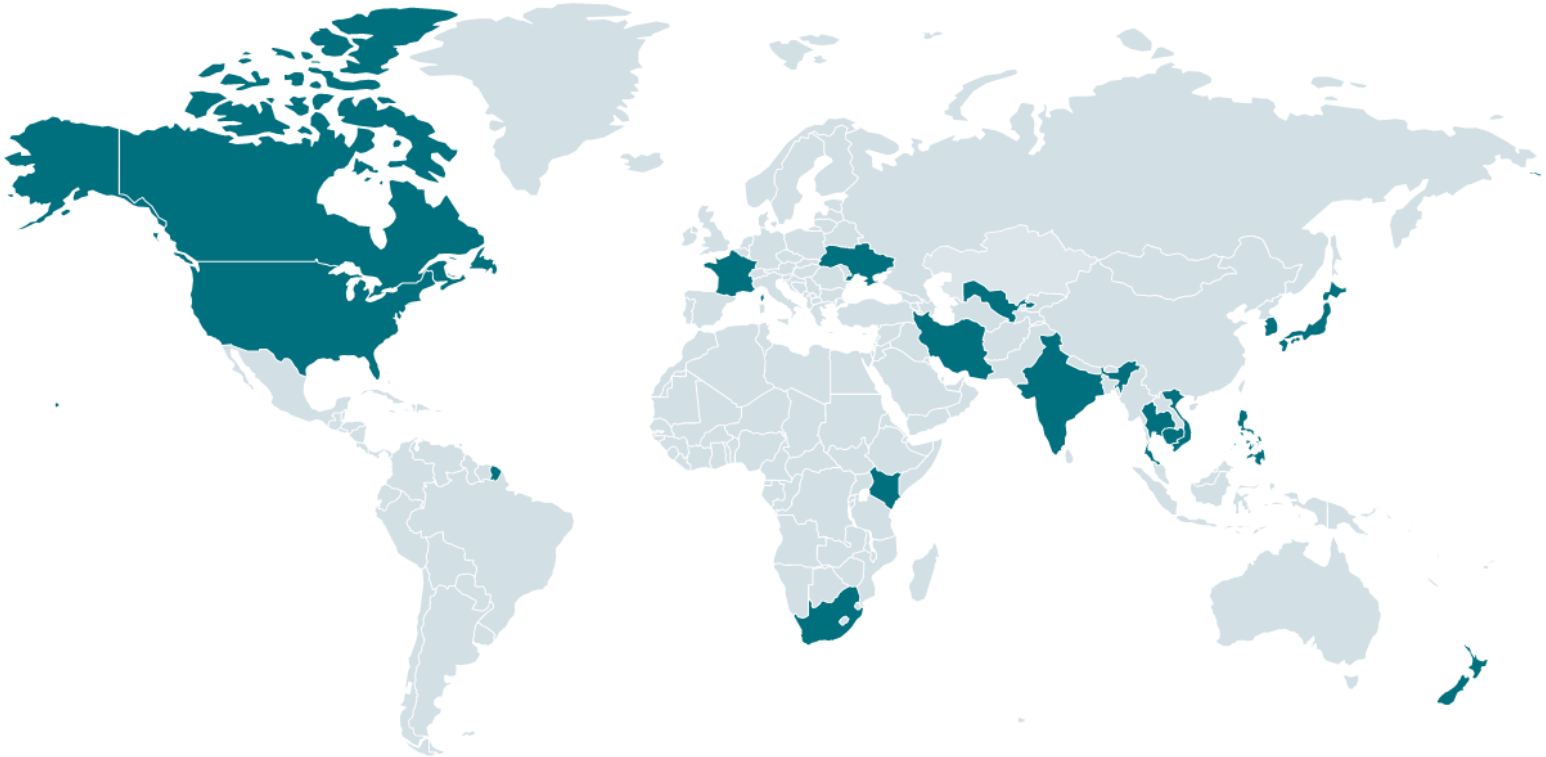
A group of IIS backdoors

spreading through ProxyLogon in 2021

Government institutions
in three countries in Southeast Asia

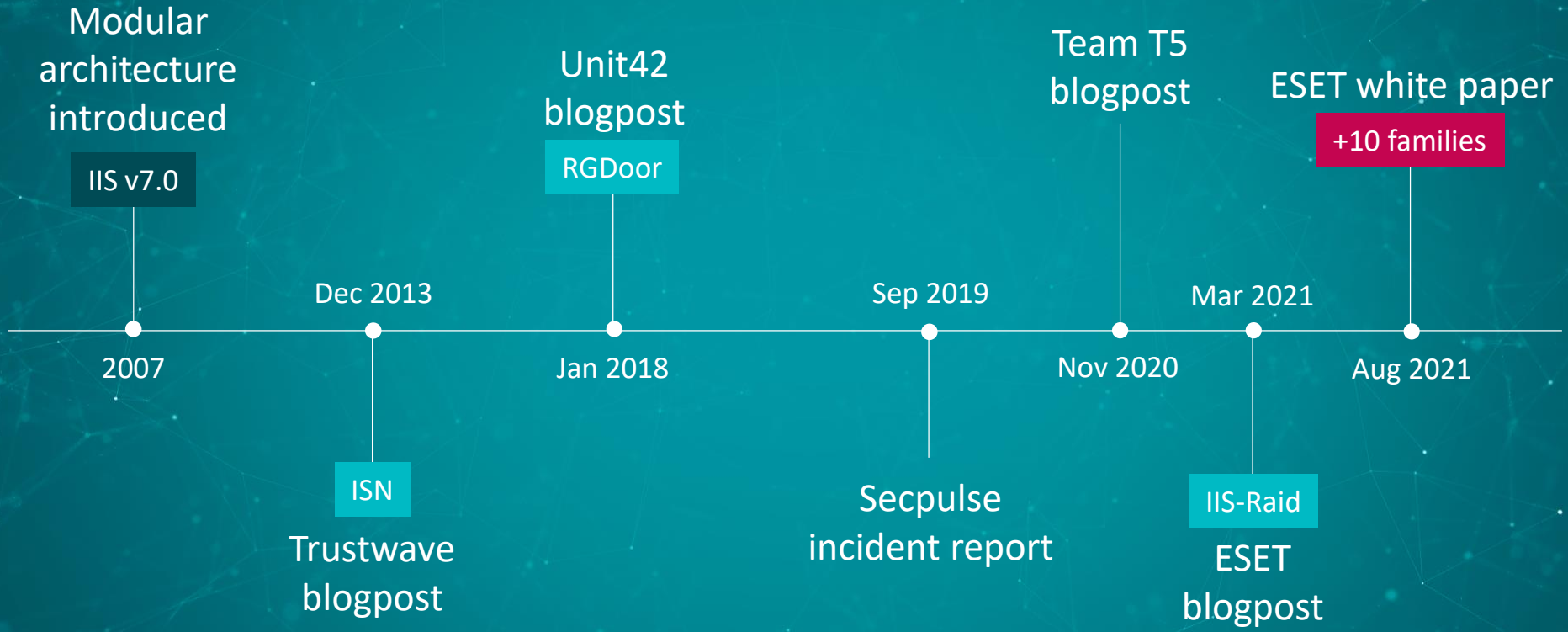
A major telecom company
in Cambodia

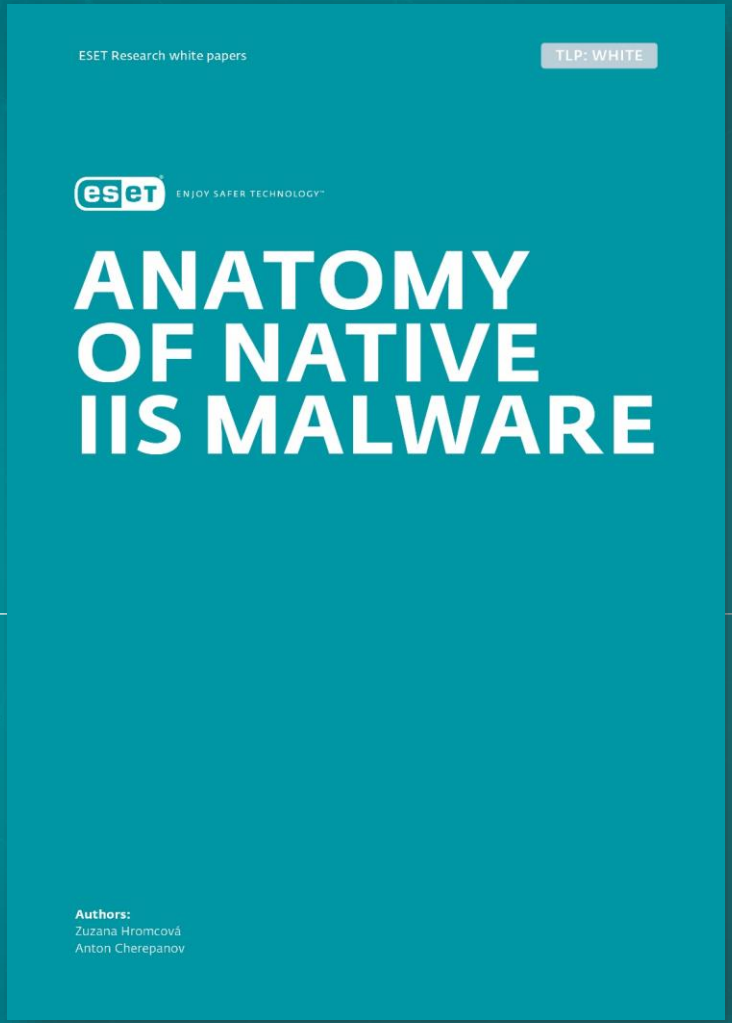
Private companies
in Canada, USA, South Korea and others



ESET ENJOY SAFER TECHNOLOGY™

IIS malware reports





ESET white paper

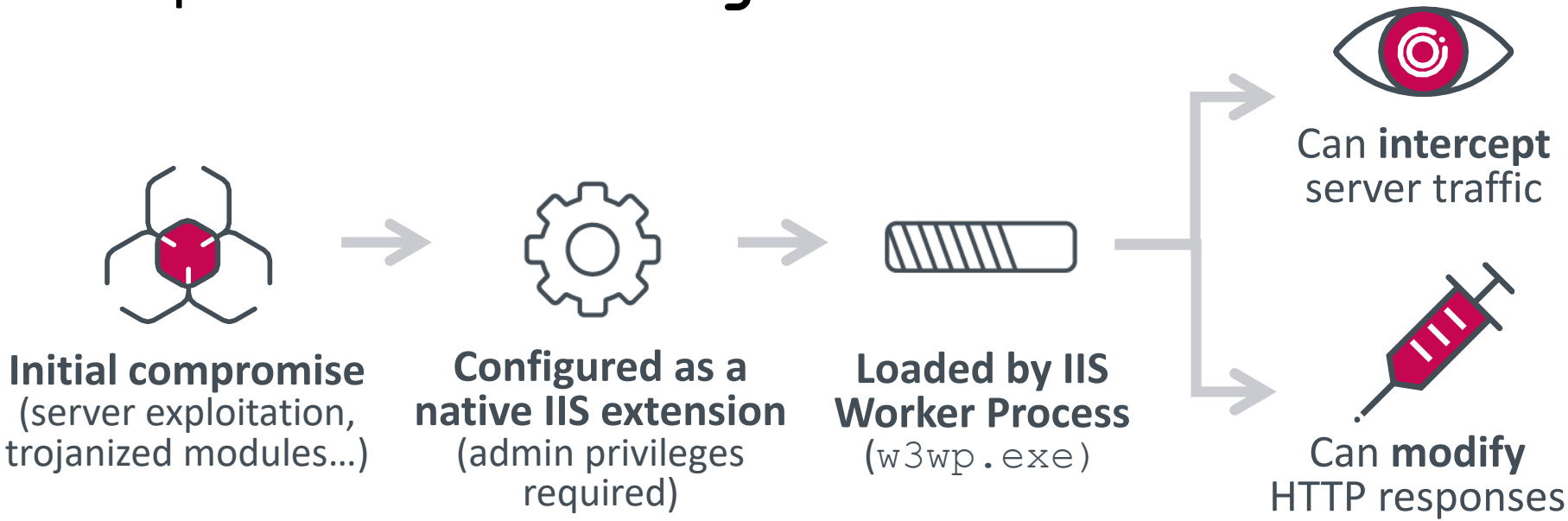
+10 families

Aug 2021

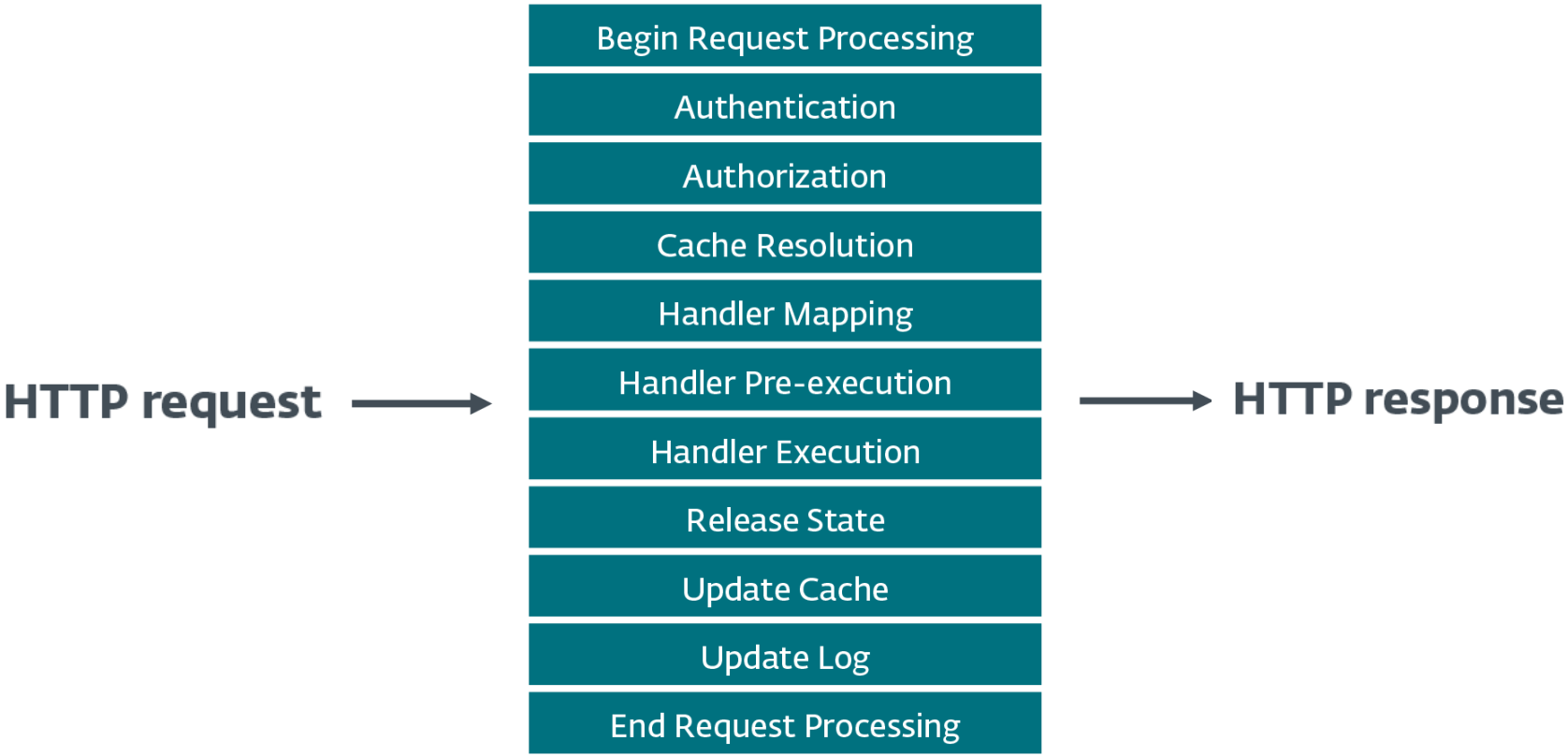
ENJOY SAFER TECHNOLOGY™

Native IIS malware

- A malicious C++ dynamic-link library
- Exports a function **`RegisterModule`**



IIS request-processing pipeline



Server events

generate notifications

- Begin Request Processing
- Authentication
- Authorization
- Cache Resolution
- Handler Mapping
- Handler Pre-execution
- Handler Execution
- Release State
- Update Cache
- Update Log
- End Request Processing

Event handlers

handle notifications

Event notification

Post-event notification

```
; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest
                    dd offset OnPostBeginRequest
                    dd offset OnAuthenticateRequest
                    dd offset OnPostAuthenticateRequest
                    dd offset OnAuthorizeRequest
                    dd offset OnPostAuthorizeRequest
                    dd offset OnResolveRequestCache
                    dd offset OnPostResolveRequestCache
                    dd offset OnMapRequestHandler
                    dd offset OnPostMapRequestHandler
                    dd offset OnAcquireRequestState
                    dd offset OnPostAcquireRequestState
                    dd offset OnPreExecuteRequestHandler
                    dd offset OnPostPreExecuteRequestHandler
                    dd offset OnExecuteRequestHandler
                    dd offset OnPostExecuteRequestHandler
                    dd offset OnReleaseRequestState
                    dd offset OnPostReleaseRequestState
                    dd offset OnUpdateRequestCache
                    dd offset OnPostUpdateRequestCache
                    dd offset OnLogRequest
                    dd offset OnPostLogRequest
```

Server events

generate notifications

- Begin Request Processing
- Authentication
- Authorization
- Cache Resolution
- Handler Mapping
- Handler Pre-execution
- Handler Execution
- Release State
- Update Cache
- Update Log
- End Request Processing

Event handlers

handle notifications

```
dd offset OnResolveRequestCache
dd offset OnPostResolveRequestCache
dd offset OnMapRequestHandler
dd offset OnPostMapRequestHandler
dd offset OnAcquireRequestState
dd offset OnPostAcquireRequestState
dd offset OnPreExecuteRequestHandler
dd offset OnPostPreExecuteRequestHandler
dd offset OnExecuteRequestHandler
dd offset OnPostExecuteRequestHandler
dd offset OnReleaseRequestState
dd offset OnPostReleaseRequestState
dd offset OnUpdateRequestCache
dd offset OnPostUpdateRequestCache
dd offset OnLogRequest
dd offset OnPostLogRequest
dd offset OnEndRequest
dd offset OnPostEndRequest
dd offset OnSendResponse
dd offset OnMapPath
dd offset OnReadEntity
dd offset OnCustomRequestNotification
dd offset OnAsyncCompletion
```

Module classes

implement event handlers

Class inheriting from **`CGlobalModule`**:

```
; const CMyGlobalModule::`vftable'
??_7CMyGlobalModule@@6B@ dq offset OnGlobalStopListening
                                        ; DATA XREF: DNameNode
                                        ; Terminate+5↑o
                dq offset OnGlobalCacheCleanup
                dq offset OnGlobalCacheOperation
                dq offset OnGlobalHealthCheck
                dq offset OnGlobalConfigurationChange
                dq offset OnGlobalFileChange
                dq offset OnGlobalPreBeginRequest
                dq offset OnGlobalApplicationStart
                dq offset OnGlobalApplicationResolveModules
                dq offset OnGlobalApplicationStop
                dq offset OnGlobalRSCAQuery
                dq offset OnGlobalTraceEvent
                dq offset OnGlobalCustomNotification
                dq offset Terminate
                dq offset OnGlobalThreadCleanup
                dq offset OnGlobalApplicationPreload
                dq offset OnSuspendProcess
```

Class inheriting from **`CHttpModule`**:

```
; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest
                                        ; DATA XREF: sub_7454A310+19↑o
                                        ; sub_7454A360+9↑o
                dd offset OnPostBeginRequest
                dd offset OnAuthenticateRequest
                dd offset OnPostAuthenticateRequest
                dd offset OnAuthorizeRequest
                dd offset OnPostAuthorizeRequest
                dd offset OnResolveRequestCache
                dd offset OnPostResolveRequestCache
                dd offset OnMapRequestHandler
                dd offset OnPostMapRequestHandler
                dd offset OnAcquireRequestState
                dd offset OnPostAcquireRequestState
                dd offset OnPreExecuteRequestHandler
                dd offset OnPostPreExecuteRequestHandler
                dd offset OnExecuteRequestHandler
                dd offset OnPostExecuteRequestHandler
                dd offset OnReleaseRequestState
                dd offset OnPostReleaseRequestState
                dd offset OnUpdateRequestCache
                dd offset OnPostUpdateRequestCache
                dd offset OnLogRequest
                dd offset OnPostLogRequest
                dd offset OnEndRequest
                dd offset OnPostEndRequest
                dd offset OnSendResponse
                dd offset OnMapPath
                dd offset OnReadEntity
                dd offset OnCustomRequestNotification
                dd offset OnAsyncCompletion
                dd offset Dispose
```

Module classes

implement event handlers

Class inheriting from **`CGlobalModule`**:

```
; const CMyGlobalModule::`vftable'
??_7CMyGlobalModule@@6B@ dq offset OnGlobalStopListening
                                        ; DATA XREF: DNameNode
                                        ; Terminate+5↑o
                dq offset OnGlobalCacheCleanup
                dq offset OnGlobalCacheOperation
                dq offset OnGlobalHealthCheck
                dq offset OnGlobalConfigurationChange
                dq offset OnGlobalFileChange
                dq offset OnGlobalPreBeginRequest
                dq offset OnGlobalApplicationStart
                dq offset OnGlobalApplicationResolveModules
                dq offset OnGlobalApplicationStop
                dq offset OnGlobalRSCAQuery
                dq offset OnGlobalTraceEvent
                dq offset OnGlobalCustomNotification
                dq offset Terminate
                dq offset OnGlobalThreadCleanup
                dq offset OnGlobalApplicationPreload
                dq offset OnSuspendProcess
```

Class inheriting from **`CHttpModule`**:

```
; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest
                                        ; DATA XREF: sub_7454A310+19↑o
                                        ; sub_7454A360+9↑o
                dd offset OnPostBeginRequest
                dd offset OnAuthenticateRequest
                dd offset OnPostAuthenticateRequest
                dd offset OnAuthorizeRequest
                dd offset OnPostAuthorizeRequest
                dd offset OnResolveRequestCache
                dd offset OnPostResolveRequestCache
                dd offset OnMapRequestHandler
                dd offset OnPostMapRequestHandler
                dd offset OnAcquireRequestState
                dd offset OnPostAcquireRequestState
                dd offset OnPreExecuteRequestHandler
                dd offset OnPostPreExecuteRequestHandler
                dd offset OnExecuteRequestHandler
                dd offset OnPostExecuteRequestHandler
                dd offset OnReleaseRequestState
                dd offset OnPostReleaseRequestState
                dd offset OnUpdateRequestCache
                dd offset OnPostUpdateRequestCache
                dd offset OnLogRequest
                dd offset OnPostLogRequest
                dd offset OnEndRequest
                dd offset OnPostEndRequest
                dd offset OnSendRespons
                dd offset OnMapPath
                dd offset OnReadEntity
                dd offset OnCustomRequestNotification
                dd offset OnAsyncCompletion
                dd offset Dispose
```

Module classes

implement event handlers

Class inheriting from **`CGlobalModule`**:

```
; const CMyGlobalModule::`vftable'
??_7CMyGlobalModule@@6B@ dq offset OnGlobalStopListening
                                        ; DATA XREF: DNameNode
                                        ; Terminate+5↑o
                          dq offset OnGlobalCacheCleanup
                          dq offset OnGlobalCacheOperation
                          dq offset OnGlobalHealthCheck
                          dq offset OnGlobalConfigurationChange
                          dq offset OnGlobalFileChange
                          dq offset OnGlobalPreBeginRequest
                          dq offset OnGlobalApplicationStart
                          dq offset OnGlobalApplicationResolveModules
                          dq offset OnGlobalApplicationStop
                          dq offset OnGlobalRSCAQuery
                          dq offset OnGlobalTraceEvent
                          dq offset OnGlobalCustomNotification
                          dq offset Terminate
                          dq offset OnGlobalThreadCleanup
                          dq offset OnGlobalApplicationPreload
                          dq offset OnSuspendProcess
```

Class inheriting from **`CHttpModule`**:

```
; const HttpModule::`vftable'
??_7CHttpModule@@6B@ dd offset OnBeginRequest
                                        ; DATA XREF: sub_7454A310+19↑o
                                        ; sub_7454A360+9↑o
                      dd offset OnPostBeginRequest
                      dd offset OnAuthenticateRequest
                      dd offset OnPostAuthenticateRequest
                      dd offset OnAuthorizeRequest
                      dd offset OnPostAuthorizeRequest
                      dd offset OnResolveRequestCache
                      dd offset OnPostResolveRequestCache
                      dd offset OnMapRequestHandler
                      dd offset OnPostMapRequestHandler
                      dd offset OnAcquireRequestState
                      dd offset OnPostAcquireRequestState
                      dd offset OnPreExecuteRequestHandler
                      dd offset OnPostPreExecuteRequestHandler
                      dd offset OnExecuteRequestHandler
                      dd offset OnPostExecuteRequestHandler
                      dd offset OnReleaseRequestState
                      dd offset OnPostReleaseRequestState
                      dd offset OnUpdateRequestCache
                      dd offset OnPostUpdateRequestCache
                      dd offset OnLogRequest
                      dd offset OnPostLogRequest
                      dd offset OnEndRequest
                      dd offset OnPostEndRequest
                      dd offset OnSendResponse
                      dd offset OnMapPath
                      dd offset OnReadEntity
                      dd offset OnCustomRequestNotification
                      dd offset OnAsyncCompletion
                      dd offset Dispose
```

RegisterModule

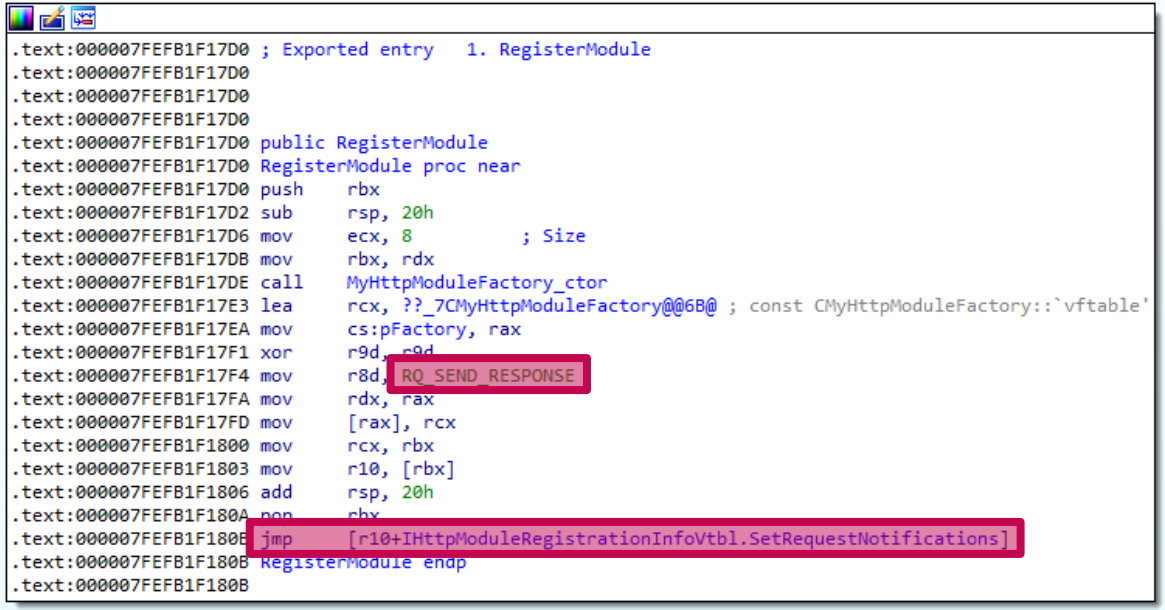
module entrypoint / DLL export

```
.text:000007FEFB1F17D0 ; Exported entry   1. RegisterModule
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0 public RegisterModule
.text:000007FEFB1F17D0 RegisterModule proc near
.text:000007FEFB1F17D0 push    rbx
.text:000007FEFB1F17D2 sub     rsp, 20h
.text:000007FEFB1F17D6 mov     ecx, 8          ; Size
.text:000007FEFB1F17DB mov     rbx, rdx
.text:000007FEFB1F17DE call    MyHttpModuleFactory_ctor
.text:000007FEFB1F17E3 lea     rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable
.text:000007FEFB1F17EA mov     cs:pFactory, rax
.text:000007FEFB1F17F1 xor     r9d, r9d
.text:000007FEFB1F17F4 mov     r8d, RQ_SEND_RESPONSE
.text:000007FEFB1F17FA mov     rdx, rax
.text:000007FEFB1F17FD mov     [rax], rcx
.text:000007FEFB1F1800 mov     rcx, rbx
.text:000007FEFB1F1803 mov     r10, [rbx]
.text:000007FEFB1F1806 add     rsp, 20h
.text:000007FEFB1F180A pop     rbx
.text:000007FEFB1F180B jmp     [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]
.text:000007FEFB1F180B RegisterModule endp
.text:000007FEFB1F180B
```

1. Creates instance of the core classes

RegisterModule

module entrypoint / DLL export

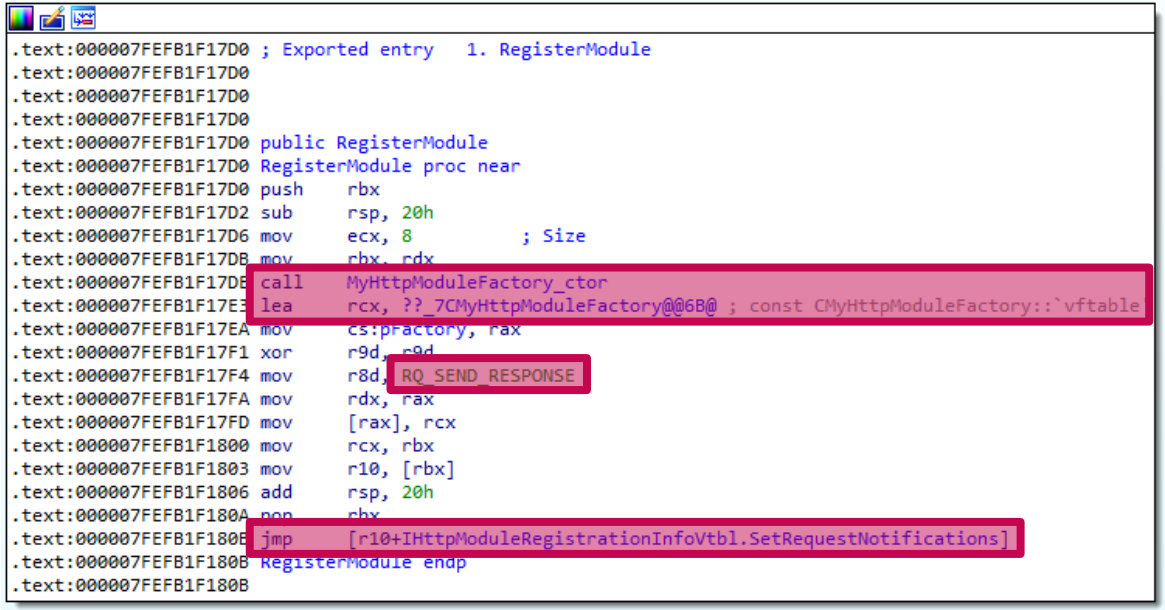


```
.text:000007FEFB1F17D0 ; Exported entry   1. RegisterModule
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0 public RegisterModule
.text:000007FEFB1F17D0 RegisterModule proc near
.text:000007FEFB1F17D0 push    rbx
.text:000007FEFB1F17D2 sub     rsp, 20h
.text:000007FEFB1F17D6 mov     ecx, 8          ; Size
.text:000007FEFB1F17DB mov     rbx, rdx
.text:000007FEFB1F17DE call    MyHttpModuleFactory_ctor
.text:000007FEFB1F17E3 lea     rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable'
.text:000007FEFB1F17EA mov     cs:pFactory, rax
.text:000007FEFB1F17F1 xor     r9d, r9d
.text:000007FEFB1F17F4 mov     r8d, RQ_SEND_RESPONSE
.text:000007FEFB1F17FA mov     rdx, rax
.text:000007FEFB1F17FD mov     [rax], rcx
.text:000007FEFB1F1800 mov     rcx, rbx
.text:000007FEFB1F1803 mov     r10, [rbx]
.text:000007FEFB1F1806 add     rsp, 20h
.text:000007FEFB1F180A pop     rbx
.text:000007FEFB1F180B jmp     [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]
.text:000007FEFB1F180B RegisterModule endp
.text:000007FEFB1F180B
```

1. Creates instance of the core classes
2. Registers module for server events

RegisterModule

module entrypoint / DLL export



```
.text:000007FEFB1F17D0 ; Exported entry   1. RegisterModule
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0 public RegisterModule
.text:000007FEFB1F17D0 RegisterModule proc near
.text:000007FEFB1F17D0 push    rbx
.text:000007FEFB1F17D2 sub     rsp, 20h
.text:000007FEFB1F17D6 mov     ecx, 8          ; Size
.text:000007FEFB1F17DB mov     rbx, rdx
.text:000007FEFB1F17DE call    MyHttpModuleFactory_ctor
.text:000007FEFB1F17E3 lea     rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable
.text:000007FEFB1F17EA mov     cs:pFactory, rax
.text:000007FEFB1F17F1 xor     r9d, r9d
.text:000007FEFB1F17F4 mov     r8d, RQ_SEND_RESPONSE
.text:000007FEFB1F17FA mov     rdx, rax
.text:000007FEFB1F17FD mov     [rax], rcx
.text:000007FEFB1F1800 mov     rcx, rbx
.text:000007FEFB1F1803 mov     r10, [rbx]
.text:000007FEFB1F1806 add     rsp, 20h
.text:000007FEFB1F180A pop     rbx
.text:000007FEFB1F180B jmp     [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]
.text:000007FEFB1F180B RegisterModule endp
.text:000007FEFB1F180B
```

1. Creates instance of the core classes
2. Registers module for server events
3. Sets priority for the module

Native IIS malware
Reversing

Reversing native IIS malware

- Identify implemented event handlers

```
.text:000007FEFB1F17D0 ; Exported entry   1. RegisterModule
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0
.text:000007FEFB1F17D0 public RegisterModule
.text:000007FEFB1F17D0 RegisterModule proc near
.text:000007FEFB1F17D0 push    rbx
.text:000007FEFB1F17D2 sub     rsp, 20h
.text:000007FEFB1F17D6 mov     ecx, 8          ; Size
.text:000007FEFB1F17DB mov     rbx, rdx
.text:000007FEFB1F17DE call    MyHttpModuleFactory_ctor
.text:000007FEFB1F17E3 lea     rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable'
.text:000007FEFB1F17EA mov     cs:pFactory, rax
.text:000007FEFB1F17F1 xor     r9d, r9d
.text:000007FEFB1F17F4 mov     r8d, RQ_SEND_RESPONSE
.text:000007FEFB1F17FA mov     rdx, rax
.text:000007FEFB1F17FD mov     [rax], rcx
.text:000007FEFB1F1800 mov     rcx, rbx
.text:000007FEFB1F1803 mov     r10, [rbx]
.text:000007FEFB1F1806 add     rsp, 20h
.text:000007FEFB1F180A pop     rbx
.text:000007FEFB1F180B jmp     [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]
.text:000007FEFB1F180B RegisterModule endp
.text:000007FEFB1F180B
```

RegisterModule

Reversing native IIS malware

- Identify implemented event handlers

```
; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest
                                        ; DATA XREF: sub_7454A310+19↑o
                                        ; sub_7454A360+9↑o
                dd offset sub_74549CD0
                dd offset sub_74549D00
                dd offset sub_74549D30
                dd offset sub_74549D60
                dd offset sub_74549D90
                dd offset sub_74549DC0
                dd offset sub_74549DF0
                dd offset sub_74549E20
                dd offset sub_74549E50
                dd offset sub_74549E80
                dd offset sub_74549EB0
                dd offset sub_74549EE0
                dd offset sub_74549F10
                dd offset sub_74549F40
                dd offset sub_74549F70
                dd offset sub_74549FA0
                dd offset sub_74549FD0
                dd offset sub_7454A000
                dd offset sub_7454A030
                dd offset OnLogRequest
                dd offset sub_7454A090
                dd offset OnEndRequest
                dd offset sub_7454A0F0
                dd offset sub_7454A120
                dd offset sub_7454A150
                dd offset sub_7454A180
                dd offset sub_7454A1B0
                dd offset sub_7454A1E0
```

default method

```
.text:00000000001417A0
.text:00000000001417A0
.text:00000000001417A0
.text:00000000001417A0 OnSendResponse proc near
.text:00000000001417A0 sub     rsp, 28h
.text:00000000001417A4 lea     rcx, OutputString ; "This module subscribed to event "
.text:00000000001417AB call    cs:OutputDebugStringA
.text:00000000001417B1 lea     rcx, aChttpmoduleOns ; "CHttpModule::OnSendResponse"
.text:00000000001417B8 call    cs:OutputDebugStringA
.text:00000000001417BE lea     rcx, aButDidNotOverr ; " but did not override the method in its"...
.text:00000000001417C5 call    cs:OutputDebugStringA
.text:00000000001417CB call    cs:DebugBreak
.text:00000000001417D1 xor     eax, eax
.text:00000000001417D3 add     rsp, 28h
.text:00000000001417D7 retn
.text:00000000001417D7 OnSendResponse endp
.text:00000000001417D7
```

Reversing native IIS malware

- Identify implemented event handlers

```
; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest
                                        ; DATA XREF: sub_7454A310+19↑o
                                        ; sub_7454A360+9↑o
                dd offset sub_74549CD0
                dd offset sub_74549D00
                dd offset sub_74549D30
                dd offset sub_74549D60
                dd offset sub_74549D90
                dd offset sub_74549DC0
                dd offset sub_74549DF0
                dd offset sub_74549E20
                dd offset sub_74549E50
                dd offset sub_74549E80
                dd offset sub_74549EB0
                dd offset sub_74549EE0
                dd offset sub_74549F10
                dd offset sub_74549F40
                dd offset sub_74549F70
                dd offset sub_74549FA0
                dd offset sub_74549FD0
                dd offset sub_7454A000
                dd offset sub_7454A030
                dd offset OnLogRequest
                dd offset sub_7454A090
                dd offset OnEndRequest
                dd offset sub_7454A0F0
                dd offset sub_7454A120
                dd offset sub_7454A150
                dd offset sub_7454A180
                dd offset sub_7454A1B0
                dd offset sub_7454A1E0
```

malicious handlers

```
1  int __thiscall OnLogRequest(httpModuleObj *this, int pHttpContext, int pProvider)
2  {
3    int pHttpRequest; // edi
4    int pHttpResponse; // eax
5
6    pHttpRequest = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetRequest)))(pHttpContext);
7    pHttpResponse = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetResponse)))(pHttpContext);
8    if ( pHttpRequest
9      && pHttpResponse
10     && ((this->flagIgnoreRequest & 1) != offsetof(httpModuleObj, field_0) || this->flagAttackerRequest) )
11   {
12     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetHttpMethod)))(pHttpRequest, "GET");
13     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetUrl)))(pHttpRequest, "/", 1, 1);
14     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Cookie");
15     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Origin");
16     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Referer");
17     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Mode");
18     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Site");
19     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Type");
20     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Length");
21     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-IP");
22     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-For");
23     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-By");
24     (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-Proto");
25   }
26   return 0;
27 }
```

Reversing native IIS malware

- Identify implemented event handlers
- Import relevant interfaces (implemented in **`iiscore.dll`**):

`IHttpContext, IHttpModuleRegistrationInfo, IHttpRequest, IHttpResponse, IPreBeginRequestProvider…`

- Refer to the [Native-Code API Reference] for the analysis

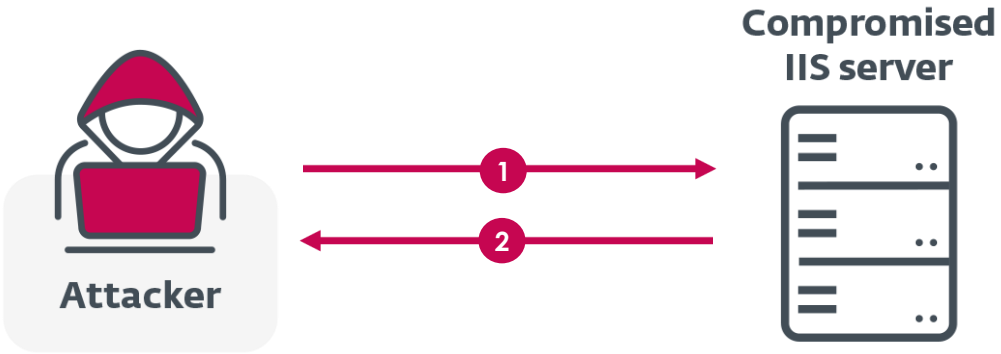
```
int __thiscall OnLogRequest(httpModuleObj *this, int pHttpContext, int pProvider)
{
  int pHttpRequest; // edi
  int pHttpResponse; // eax

  pHttpRequest = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetRequest)))(pHttpContext);
  pHttpResponse = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetResponse)))(pHttpContext);
  if ( pHttpRequest
    && pHttpResponse
    && ((this->flagIgnoreRequest & 1) != offsetof(httpModuleObj, field_0) || this->flagAttackerRequest) )
  {
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetHttpMethod)))(pHttpRequest, "GET");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetUrl)))(pHttpRequest, "/", 1, 1);
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Cookie");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Origin");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Referer");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Mode");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Site");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Type");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Length");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-IP");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-For");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-By");
    (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-Proto");
  }
  return 0;
}
```

Native IIS malware

Malware types

#1 IIS backdoors

execute backdoor commands on IIS server

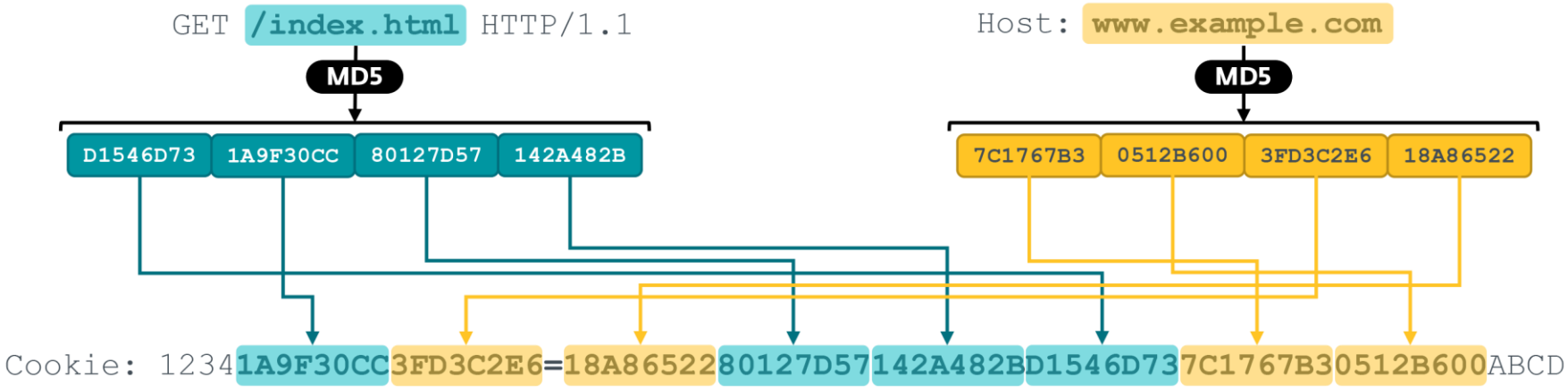


Backdoor commands

- Get system information
- Upload/download files
- Execute files or shell commands
- Create reverse shell
- Create/list/move/rename/delete files and folders
- Map local drives to remote drives
- Exfiltrate collected data

#1 IIS backdoors

execute backdoor commands on IIS server

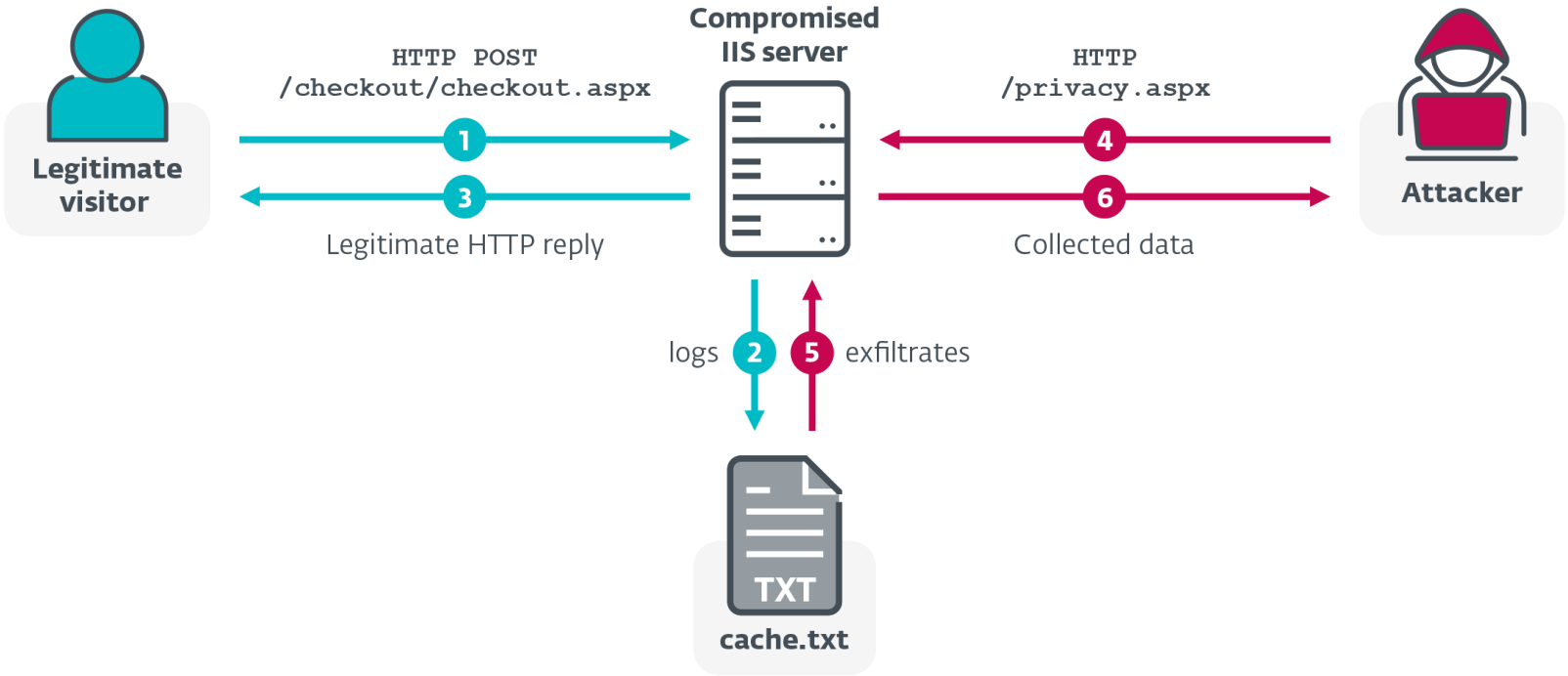


Control HTTP requests

- A custom HTTP header present
- A specific format of URL or request body
- An embedded password in the URL, request body, headers (hardcoded password or password hash in the malware)
- A more complex condition

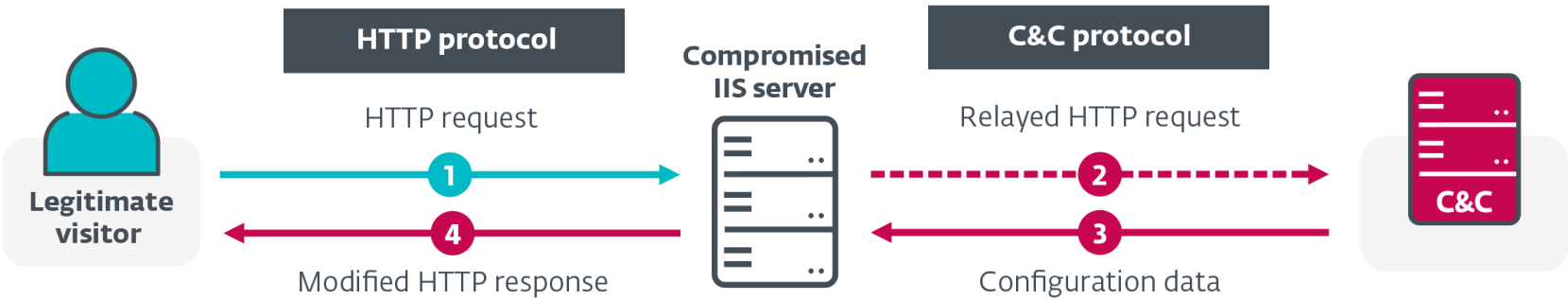
#2 IIS infostealers

intercept traffic and steal data from legitimate visitors



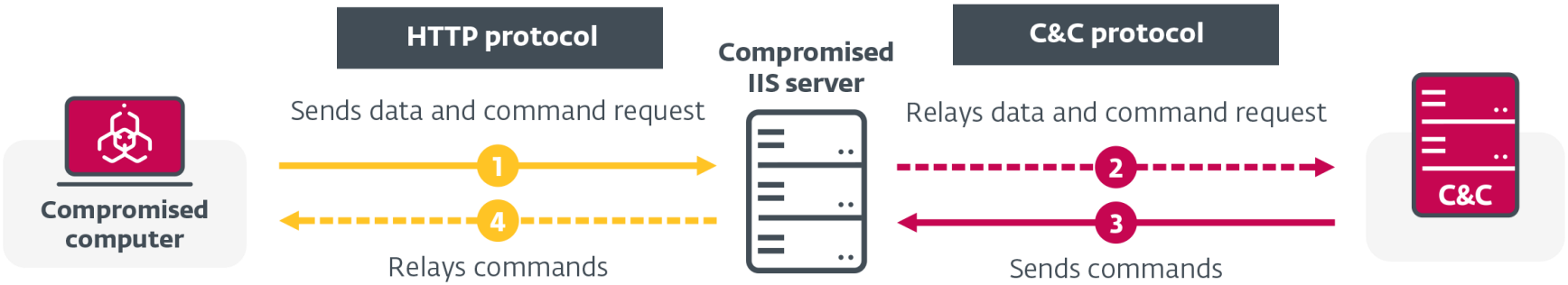
#3 IIS injectors

serve malicious content to legitimate visitors



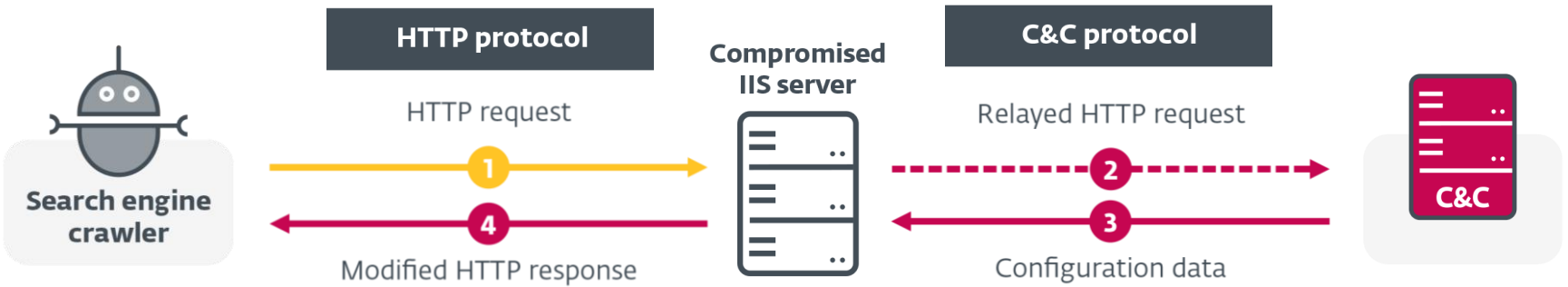
#4 IIS proxies

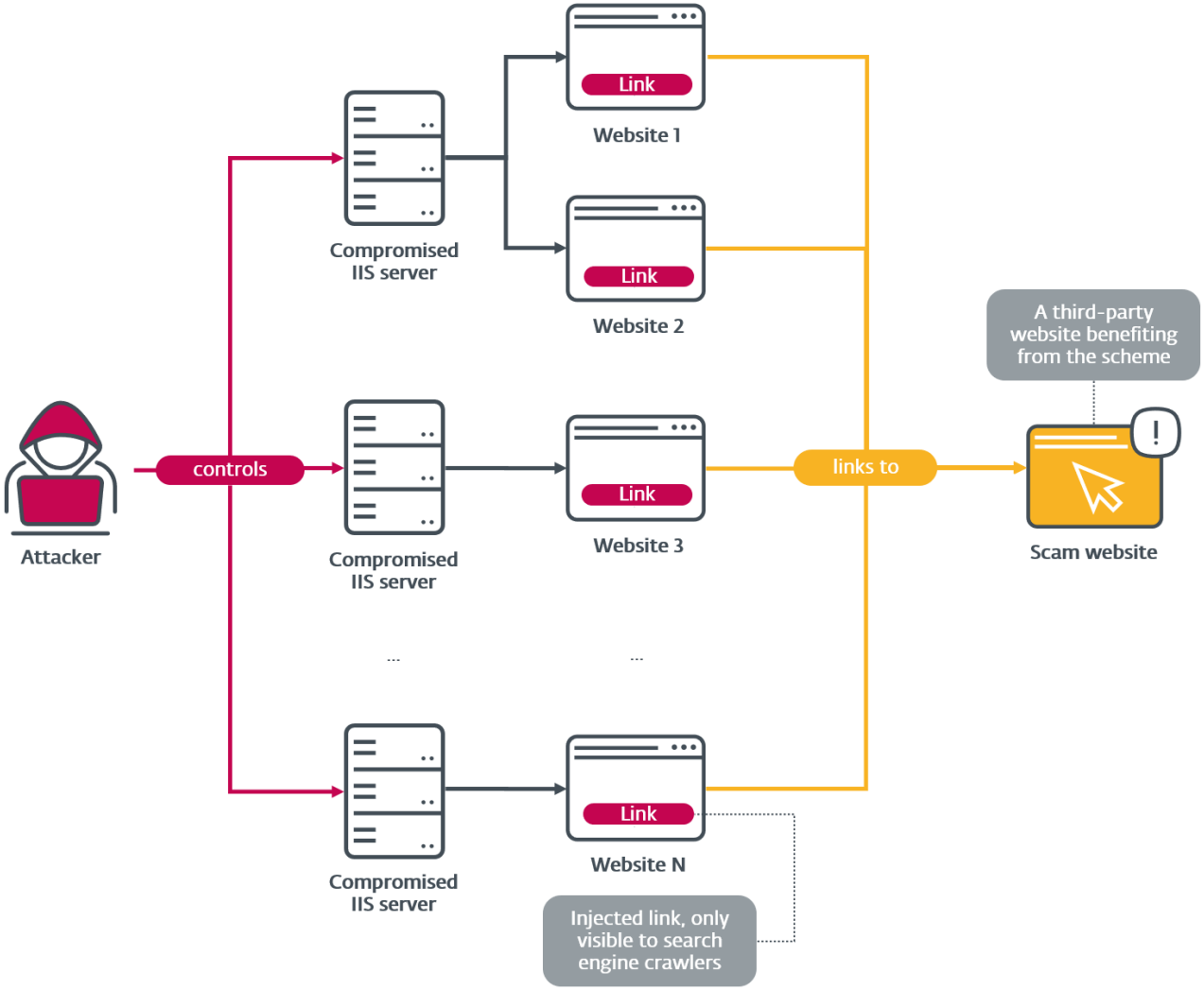
relay traffic between a compromised host and the C&C server



#5 SEO fraud malware

manipulates content served to search engine crawlers to boost SEO of third-party websites





Known IIS malware families

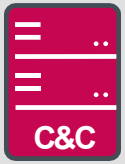
see our paper for detailed analyses

Malware family	Backdoor	Info stealer	Proxy	SEO fraud	Injector
Group 1 (IIS-Raid)	✓	✓			
Group 2	✓				
Group 3	✓				
Group 4 (RGDoor)	✓				
Group 5 (IIStealer)		✓			
Group 6 (ISN)		✓			
Group 7 (IISpy)	✓				
Group 8	✓				
Group 9			✓	✓	
Group 10				✓	
Group 11	✓		✓	✓	✓
Group 12A	✓		✓	✓	✓
Group 12B	✓			✓	✓
Group 12C				✓	
Group 13 (IISerpent)	✓			✓	
Group 14				✓	✓

Conclusion

Built-in **persistence as IIS extension**

Loaded by IIS Worker Process `w3wp.exe`



Built-in **passive C&C channel**

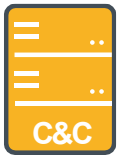
Can **intercept** server traffic

Can **modify** HTTP responses

Native IIS malware features

Targets government mailboxes (IIS backdoors)

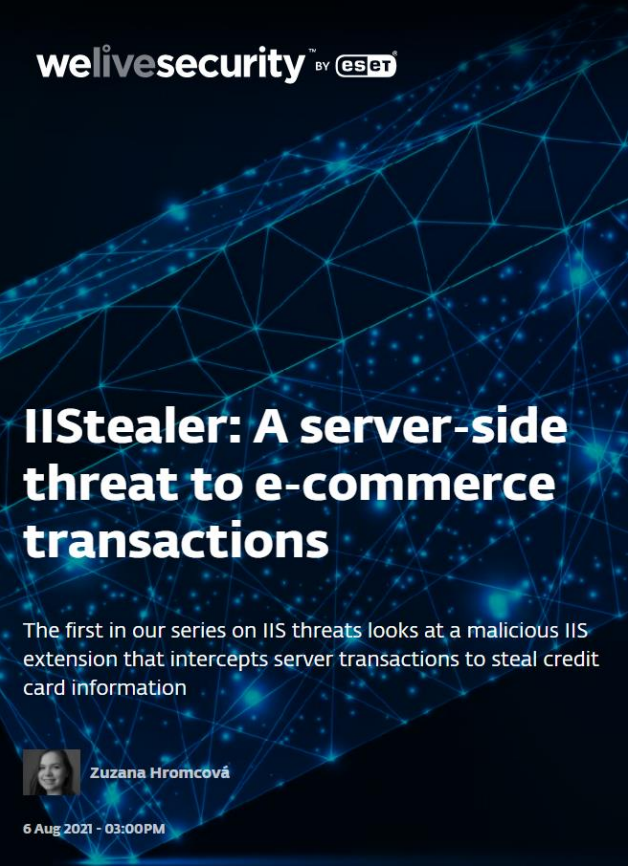
Targets e-commerce websites (IIS infostealers)

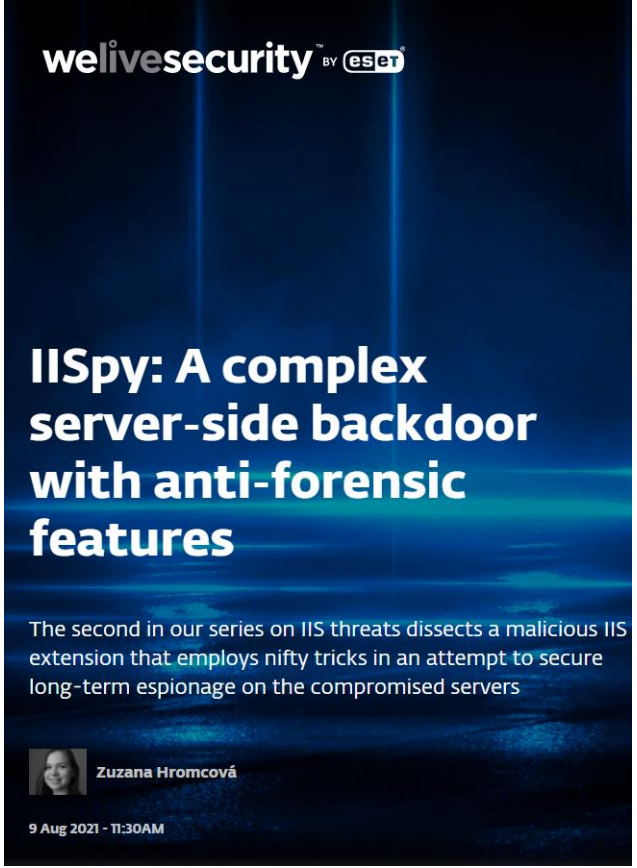


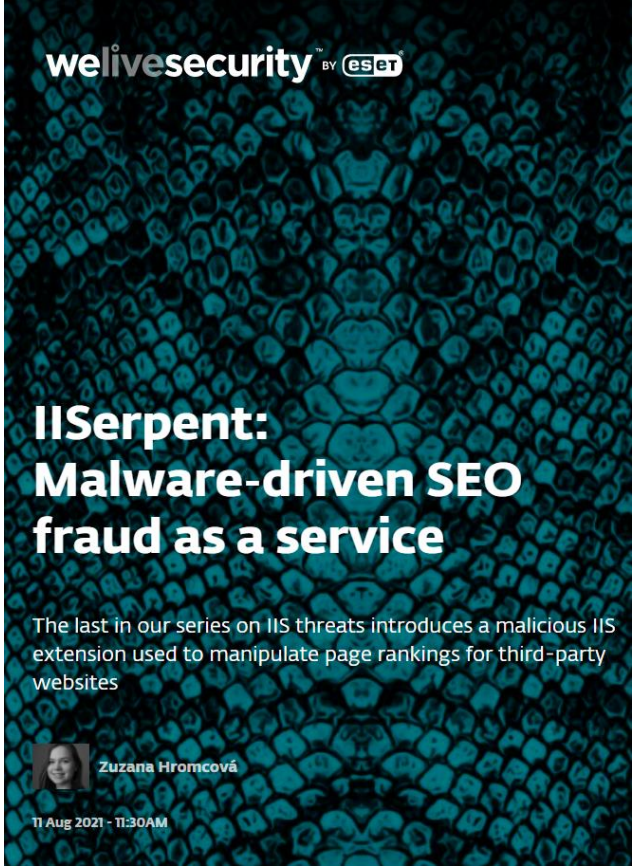
Aids in C&C routing (IIS proxies)

Manipulates SERP (SEO fraud)

Aids in malware distribution (IIS injectors)







Read our blogpost series

IIS threats under the microscope

Read the white paper
in the conference
proceedings

Zuzana Hromcova

ESET Malware Researcher

@zuzana_hromcova

Anton Cherepanov

ESET Senior Malware Researcher

@cherepanov74

ENJOY SAFER TECHNOLOGY™

THANK YOU!

www.eset.com | www.welivesecurity.com | @ESETresearch