Anatomy of Native IIS Malware

Zuzana Hromcova | Malware Researcher, ESET

Anatomy of Native IIS Malware

C++ libraries

Internet Information Services

How popular is IIS software?

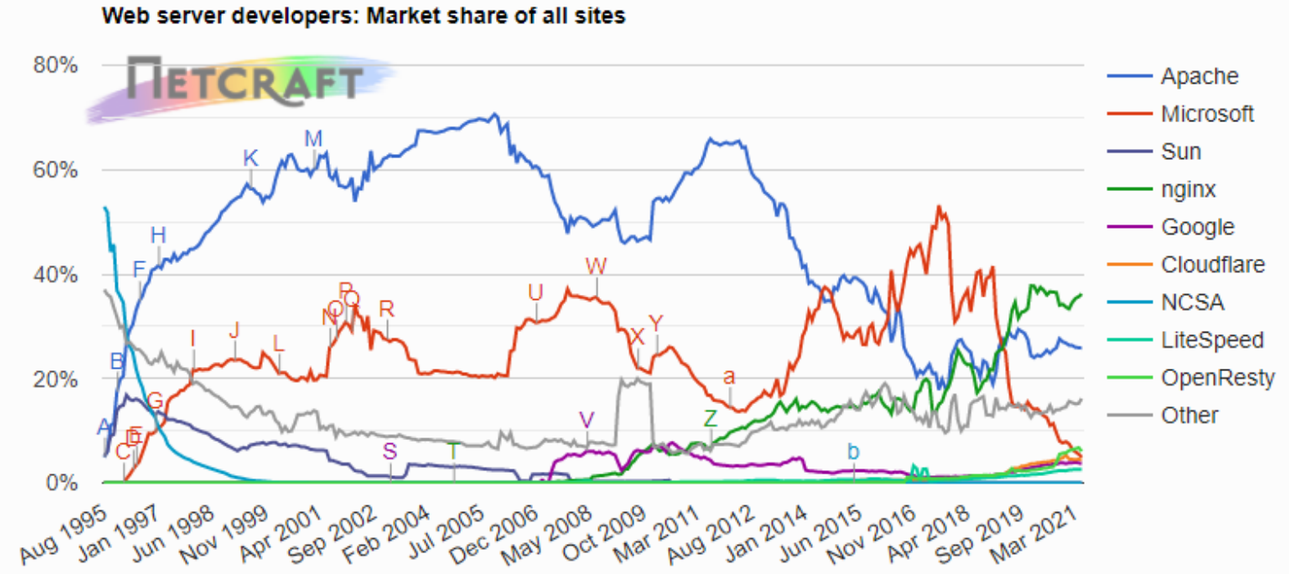
4-7%

of websites use IIS server*

4-7%

of websites use IIS server*

*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 server*

*W3Techs: Usage statistics of web servers

Technologies > Web Servers

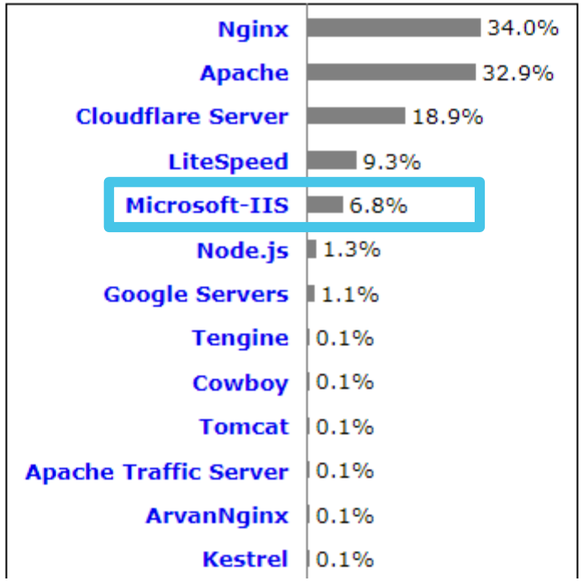
Usage statistics of web servers

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.



Microsoft Exchange email servers with

OWA

Outlook on the web

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):

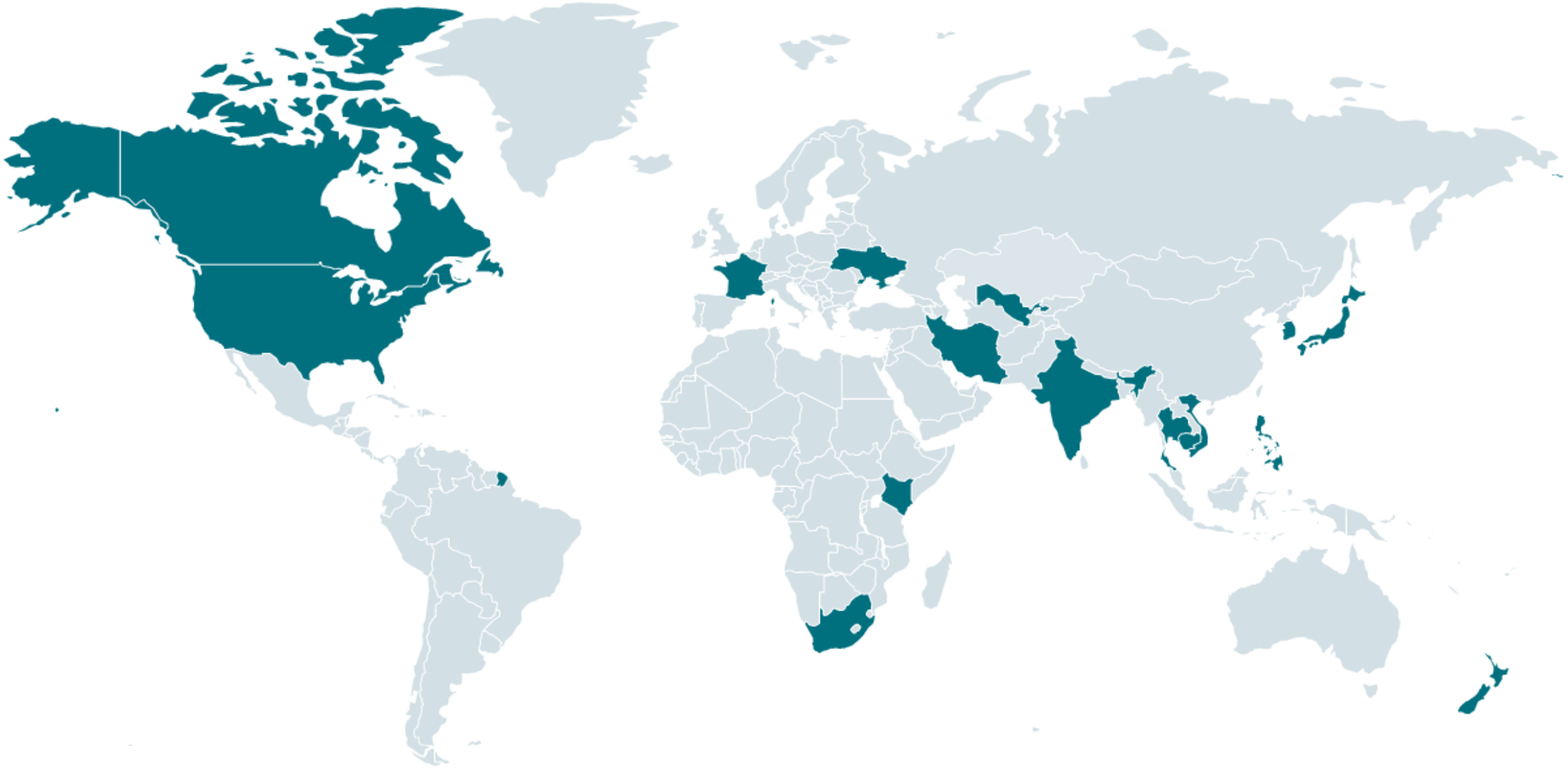
IIS backdoors

spreading via ProxyLogon

Government institutions
in three countries in Southeast Asia

A major telecom company
in Cambodia

Private companies
in Canada, USA, South Korea and others



Government espionage

Infiltrating government mailboxes

SEO fraud

Crime schemes to manipulate SERP

Compromised websites

IIS malware serving malicious content & adware

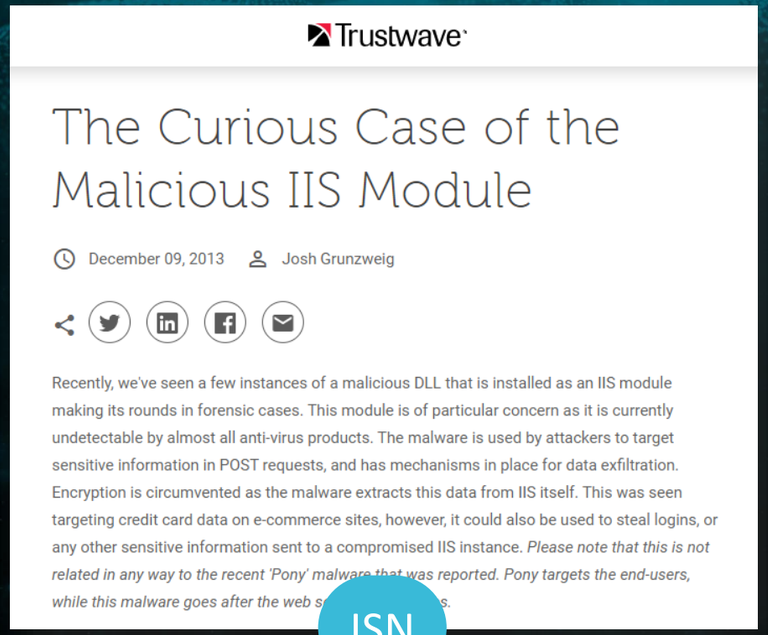
Targeting e-commerce

Stealing credentials & credit card information

C&C traffic routing

Compromised IIS server as a malicious proxy

Known malicious IIS modules:



Trustwave

The Curious Case of the Malicious IIS Module

December 09, 2013 Josh Grunzweig

Recently, we've seen a few instances of a malicious DLL that is installed as an IIS module making its rounds in forensic cases. This module is of particular concern as it is currently undetectable by almost all anti-virus products. The malware is used by attackers to target sensitive information in POST requests, and has mechanisms in place for data exfiltration. Encryption is circumvented as the malware extracts this data from IIS itself. This was seen targeting credit card data on e-commerce sites, however, it could also be used to steal logins, or any other sensitive information sent to a compromised IIS instance. *Please note that this is not related in any way to the recent 'Pony' malware that was reported. Pony targets the end-users, while this malware goes after the web s[illegible]s.*

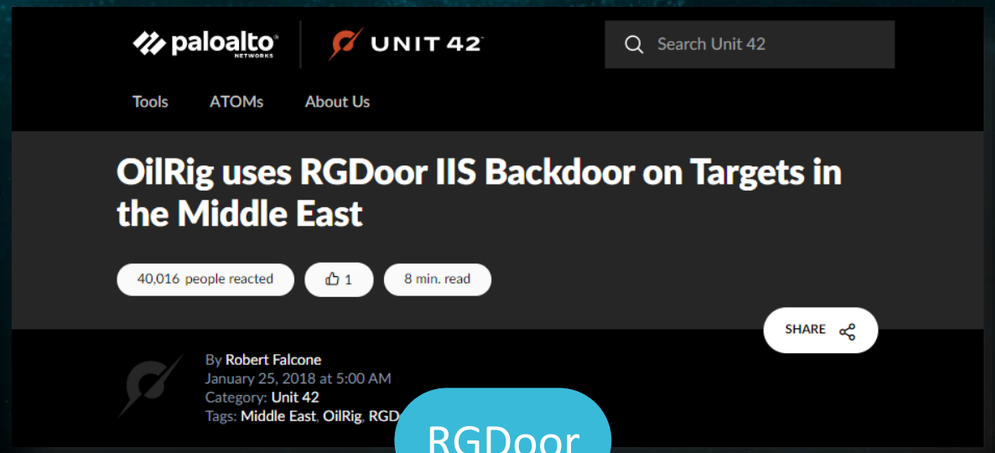
ISN

IIS v7.0

2007

2013

Known malicious IIS modules:



IIS v7.0

2007

ISN

2013

RGDoor

2018

Known malicious IIS modules:

IIS v7.0 — 2007

ISN — 2013

Native malware — 2018 / 2019

Known malicious IIS modules:

IIS v7.0 — 2007

ISN — 2013

RGDoor — 2018

Native malware — 2019

Managed malware — 2020

Known malicious IIS modules:

IIS v7.0 — 2007

ISN — 2013

RGDoor — 2018

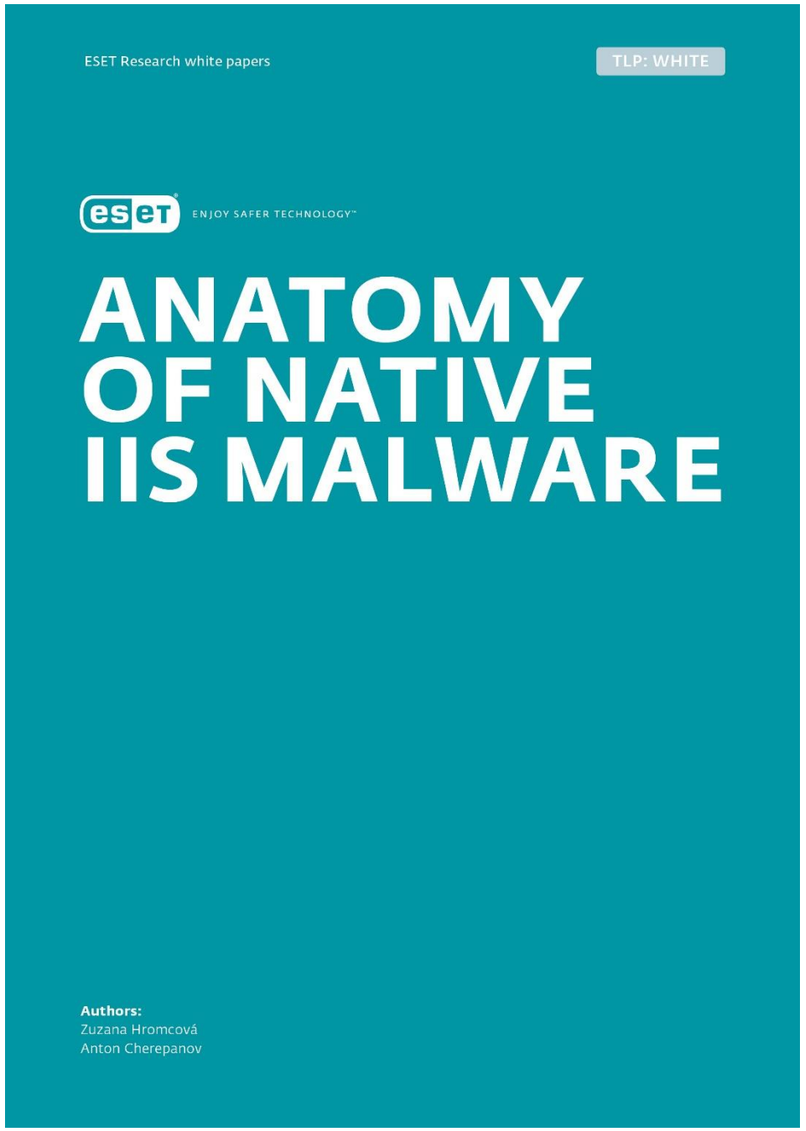
Native malware — 2019

2020

IIS-Raid — 2021

Known malicious IIS modules:

Our research



Our research

Malicious **native** IIS modules (C++ libraries)

80+ unique samples from our telemetry and VirusTotal

14 malware families
(10 never documented)

Victim information from our telemetry and internet-wide scans

Detailed information and analyses in the white paper

Malicious native IIS modules

Architecture
Reversing
TTPs
Defense

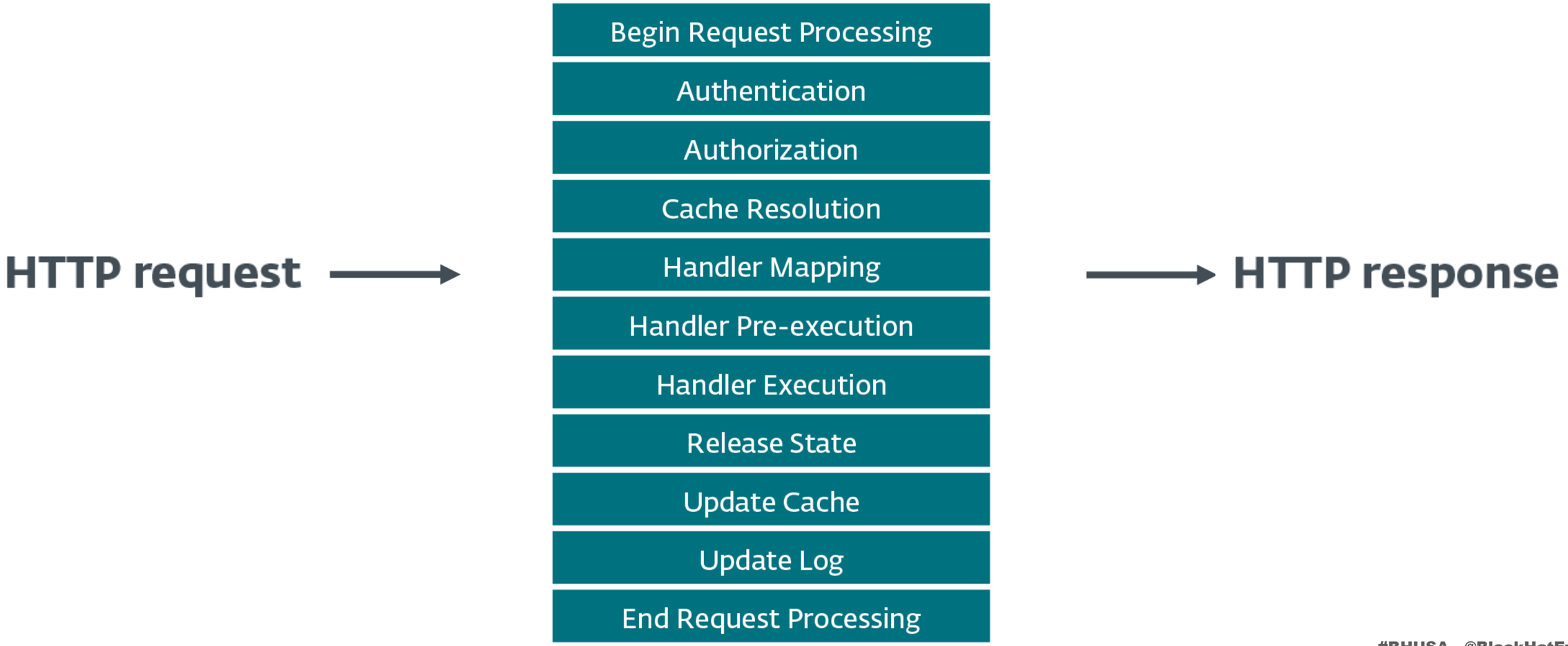
Malicious native IIS modules

Architecture

Internet Information Services (IIS)

- Microsoft web server software
- Modular architecture (since v7.0)
- IIS services configured to run at each system start (*World Wide Web Publishing Service, Windows Process Activation Service or Application Host Helper Services*)
- IIS Worker Process (**`w3wp.exe`**)
  - Handles inbound requests
  - Loads all IIS modules configured in **`%windir%\system32\inetsrv\config\ApplicationHost.config`**

Request-processing pipeline



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 notification

Post-event notification

Event handlers

handle notifications

Class inheriting from `CHttpModule`:

```
; 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
```

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

```
; 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
                    dd offset OnEndRequest
                    dd offset OnPostEndRequest
                    dd offset OnSendResponse
                    dd offset OnMapPath
```

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 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 OnSendResponse
                dd offset OnMapPath
                dd offset OnReadEntity
                dd offset OnCustomRequestNotification
                dd offset OnAsyncCompletion
                dd offset Dispose
```

RegisterModule

DLL export / module entrypoint

1. Creates instances of the core classes

```
.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

DLL export / module entrypoint

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

```
.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

DLL export / module entrypoint

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

```
.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
```

Malicious native IIS modules

Reverse-engineering

1 Import relevant interfaces (implemented in **`iiscore.dll`**):

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

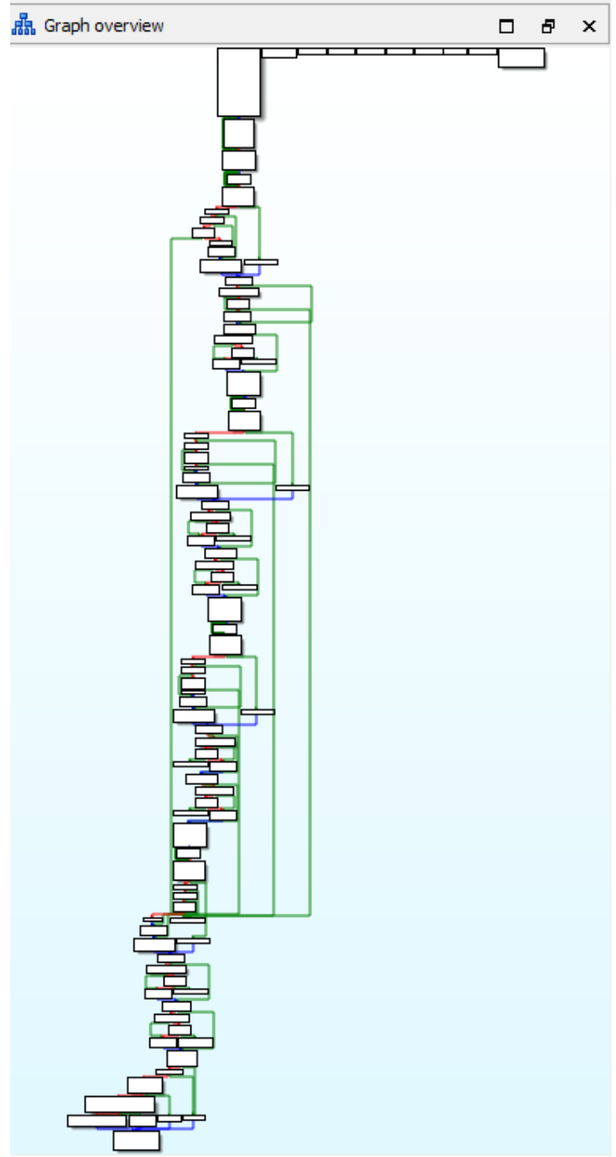
```
1  int __cdecl sendResponse(const char *a1, int cHttpContext)
2  {
3    IHttpResponse **cHttpResponse1; // eax
4    IHttpResponse **cHttpResponse2; // esi
5    IHttpResponse *cHttpResponse3; // edx
6    int hResult1; // eax
7    int hResult2; // edi
8    HTTP_DATA_CHUNK httpDataChunks; // [esp+4h] [ebp-40h] BYREF
9
10   cHttpResponse1 = (*(*cHttpContext + offsetof(IHttpContext2Vtbl, GetResponse)))(cHttpContext);
11   cHttpResponse2 = cHttpResponse1;
12   if ( !cHttpResponse1 )
13     return 1;
14   cHttpResponse3 = *cHttpResponse1;
15   httpDataChunks.DataChunkType = HttpDataChunkFromMemory;
16   (cHttpResponse3->Clear)(cHttpResponse1);
17   ((*cHttpResponse2)->SetHeader)(cHttpResponse2, HttpHeaderContentType, "text/plain", 10, 1);
18   httpDataChunks.FromFileHandle.ByteRange.StartingOffset.LowPart = a1;
19   httpDataChunks.FromMemory.BufferLength = strlen(a1);
20   hResult1 = ((*cHttpResponse2)->WriteEntityChunks)(cHttpResponse2, &httpDataChunks, 1, 0, 1, &cHttpContext, 0);
21   hResult2 = hResult1;
22   if ( hResult1 < 0 )
23     ((*cHttpResponse2)->SetStatus)(cHttpResponse2, 500, "Server Error", 0, hResult1, 0, 0);
24   return hResult2;
25 }
```

2 Start with `RegisterModule` export

- Which handlers are implemented?
- Initialization?

```
.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
```

See Group 9 in the paper



3 Identify implemented handlers

```
.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
```

See Group 7 in the paper

```
; 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
                    dd offset sub_7454A210
                    dd offset sub_7454A360
                    dd offset ??_R4HttpModuleFactory@@6B@ ; const HttpModuleFact
; const HttpModuleFactory::`vftable'
```

malicious handler

malicious handler

malicious handler

3 Identify implemented handlers

```
.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
```

See Group 12 in the paper

```
; const CF5XFFHttpModule::`vftable'
??_7CF5XFFHttpModule@@6B@ dq offset OnBeginRequest
                                        ; DATA XREF: sub_180005900+A↑o
                                        ; sub_1800059C0+4D↑o
                dq offset sub_180005DF0
                dq offset sub_180005AF0
                dq offset sub_180005D70
                dq offset sub_180005B30
                dq offset sub_180005DB0
                dq offset sub_1800060F0
                dq offset sub_180005FB0
                dq offset sub_180005CF0
                dq offset sub_180005FE0
                dq offset OnAcquireRequestState
                dq offset sub_180005D30
                dq offset sub_180006030
                dq offset sub_180005F30
                dq offset sub_180005C30
                dq offset sub_180005E70
                dq offset sub_1800060B0
                dq offset sub_180005F70
                dq offset sub_180006170
                dq offset sub_180005FF0
                dq offset sub_180005C70
                dq offset sub_180005EB0
                dq offset sub_180005BF0
                dq offset sub_180005F30
                dq offset OnSendResponse
                dq offset sub_180005CB0
                dq offset sub_180006070
                dq offset sub_180005BB0
                dq offset sub_180005AB0
                dq offset sub_1800059A0
                dq offset sub_180005900
```

malicious handler

benign handler

benign handler

3 Identify implemented handlers

```
.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
```

See Group 12 in the paper

```
; const CF5XFFHttpModule::`vftable'
??_7CF5XFFHttpModule@@6B@ dq offset OnBeginRequest
                                        ; DATA XREF: sub_180005900+A↑o
                                        ; sub_1800059C0+4D↑o
                dq offset sub_180005DF0
                dq offset sub_180005AF0
                dq offset sub_180005D70
                dq offset sub_180005B30
                dq offset sub_180005DB0
                dq offset sub_1800060F0
                dq offset sub_180005FB0
                dq offset sub_180005CF0
                dq offset sub_180005FE0
                dq offset OnAcquireRequestState
                dq offset sub_180005D30
                dq offset sub_180006030
                dq offset sub_180005F30
                dq offset sub_180005C30
                dq offset sub_180005E70
                dq offset sub_1800060B0
                dq offset sub_180005F70
                dq offset sub_180006170
                dq offset sub_180005FF0
                dq offset sub_180005C70
                dq offset sub_180005EB0
                dq offset sub_180005BF0
                dq offset sub_180005F30
                dq offset OnSendResponse
                dq offset sub_180005CB0
                dq offset sub_180006070
                dq offset sub_180005BB0
                dq offset sub_180005AB0
                dq offset sub_1800059A0
                dq offset sub_180005900
```

malicious handler

benign handler

benign handler

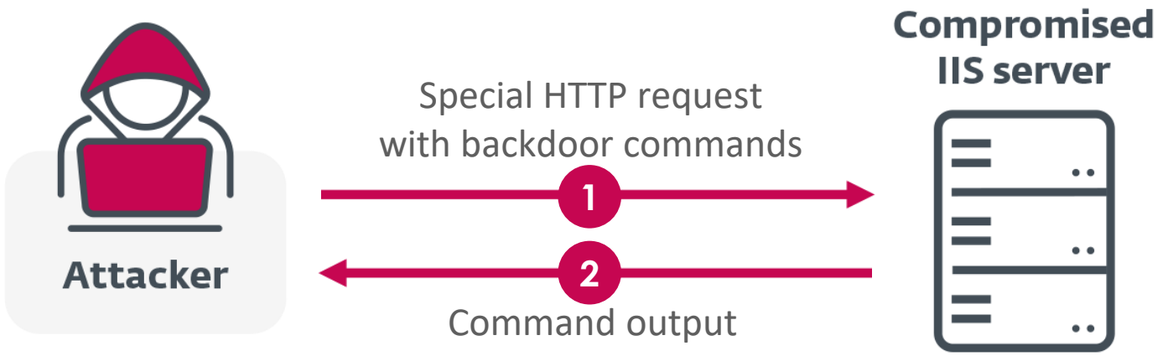
4 Refer to the Native-Code API Reference for the analysis

Malicious native IIS modules

Understanding the TTPs

1 IIS backdoors

execute backdoor commands on IIS server



Backdoor commands

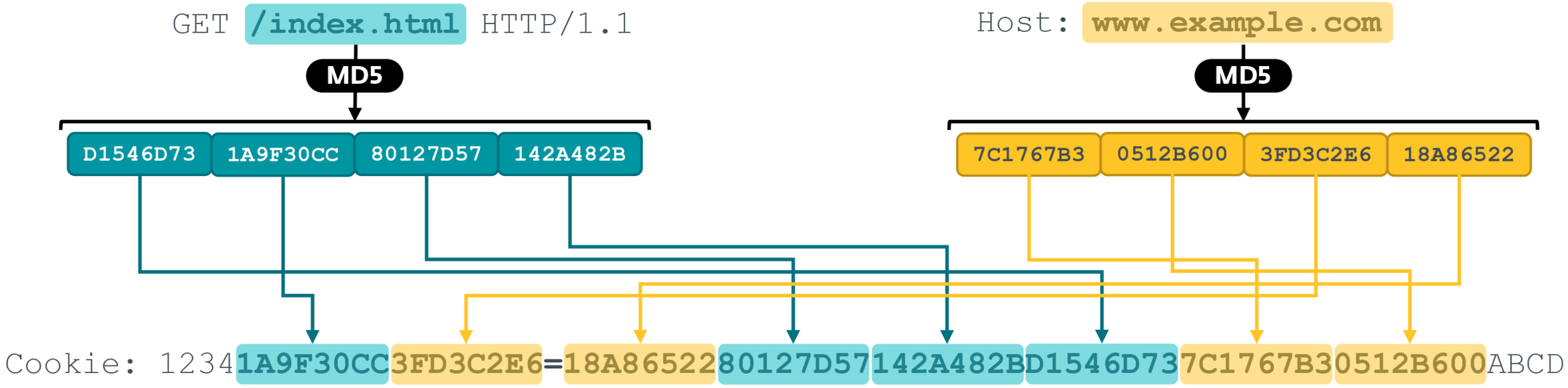
- Get system information
- Upload/download files
- Execute files or shell commands
- Create reverse shell
- Create/list/move/rename/delete files and folders
- Map local drives to remote drives
- Exfiltrate collected data

Attacker HTTP requests

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

Attacker HTTP request example

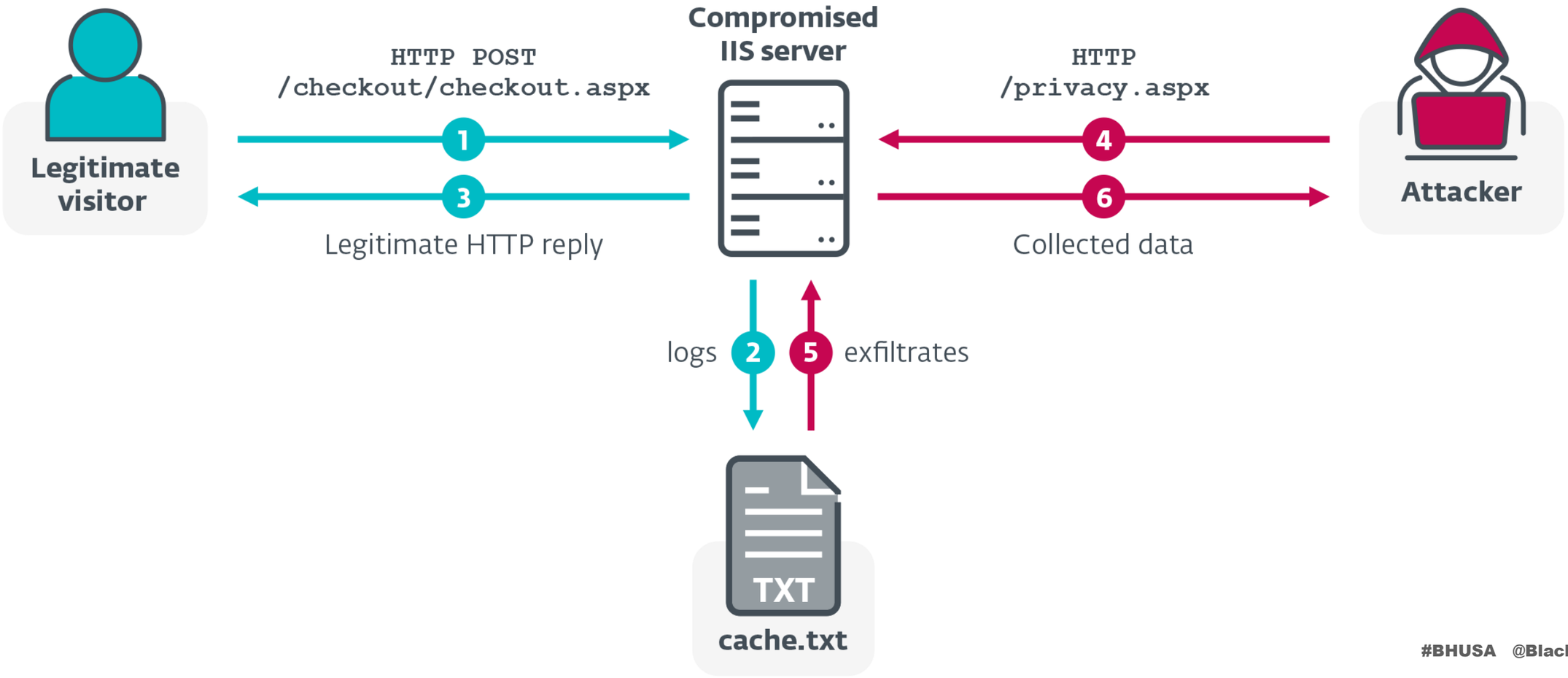
See Group 7 in the paper



See Group 5 in the paper

2 IIS infostealers

intercept traffic and steal data from legitimate visitors

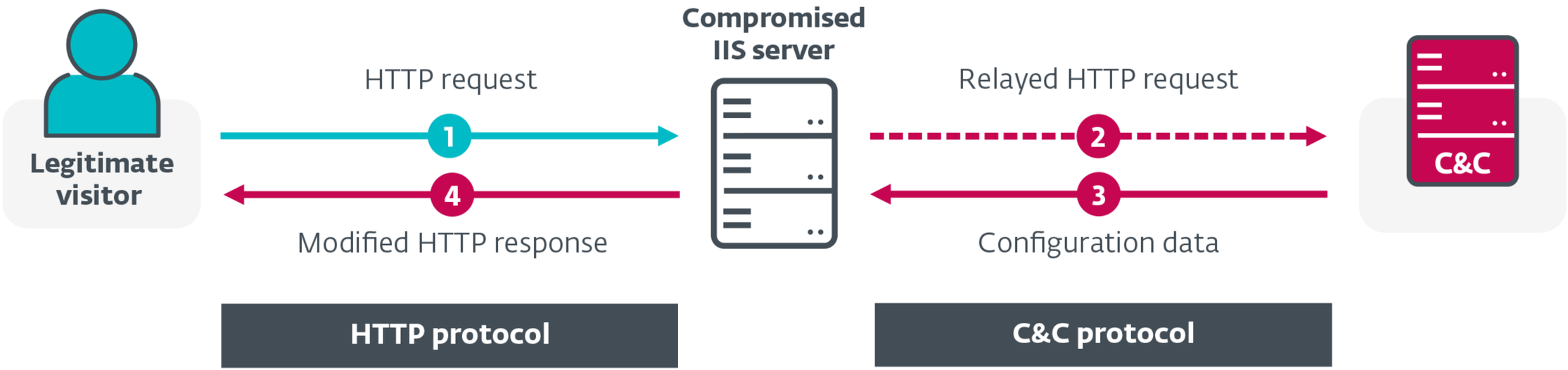


DEMO

See Group 12 in the paper

3 IIS injectors

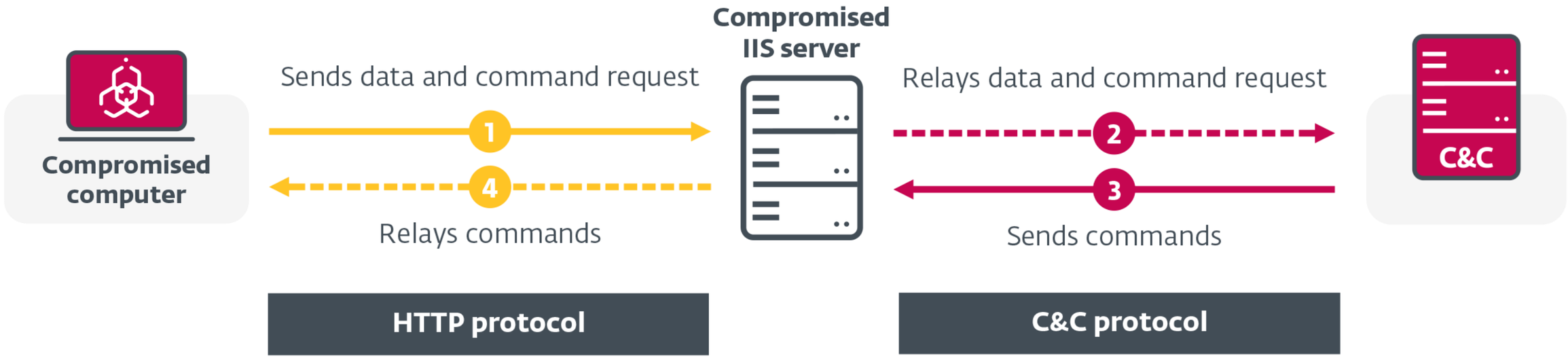
serve malicious content to legitimate visitors



See Group 9 in the paper

4 IIS proxies

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



See Group 13 in the paper

5 SEO fraud

deceive search engine crawlers

- Manipulates content served to **search engine crawlers** to boost SEO for selected websites
- Legitimate user requests are ignored by the malware
- Techniques used:
  - Keyword stuffing
  - Injecting a list of backlinks
  - Redirecting the crawlers (turning the compromised website into a doorway page)
- This is **not** Black Hat SEO
  - A third-party website benefits from the manipulation, not the one serving the manipulated content (this is likely sold as a service)
  - C&C communication to obtain configuration data
  - Other malicious modes present (e.g. backdoor, proxy)

Known IIS malware families

See the paper for detailed analyses

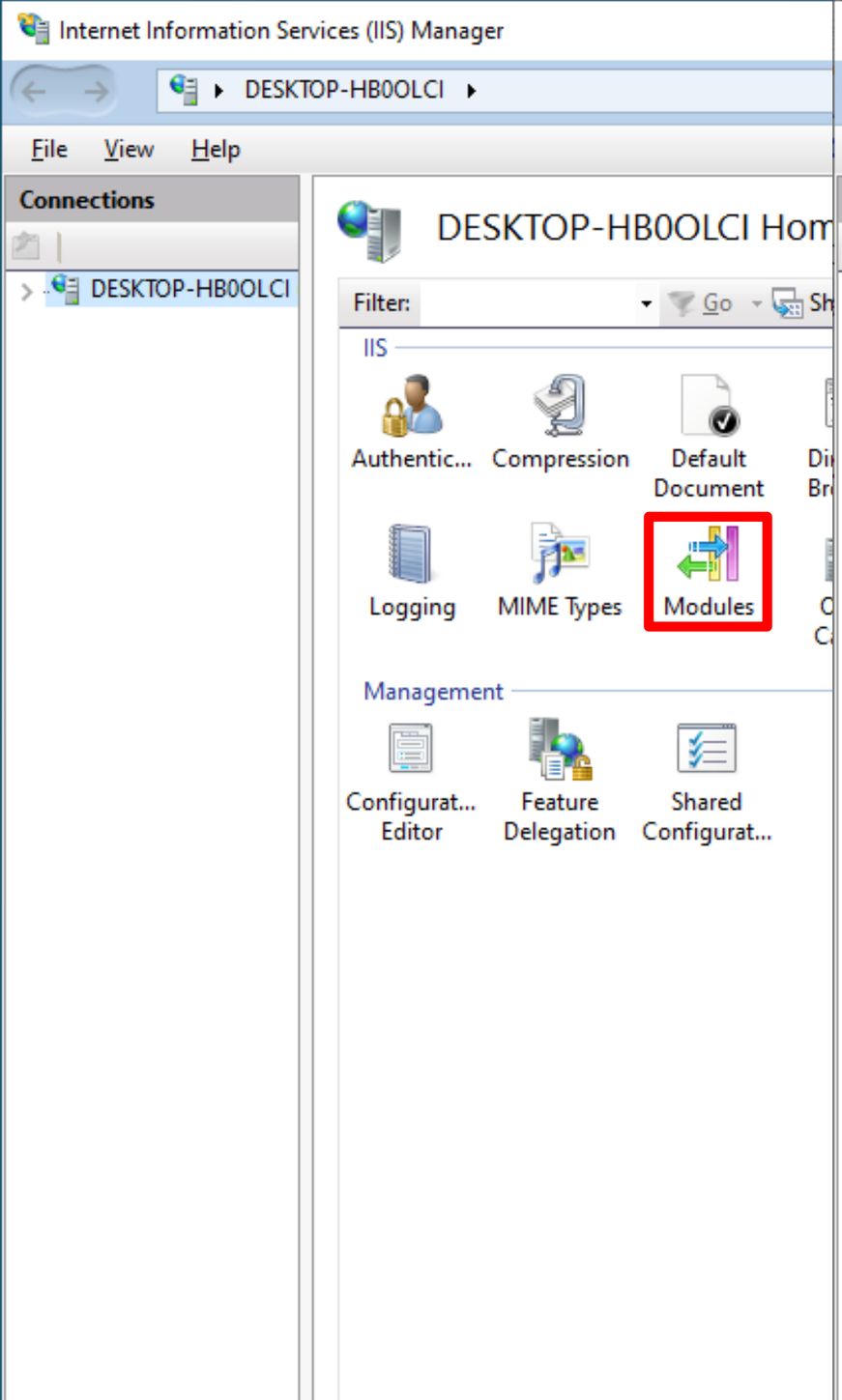
Malware family	Backdoor	Info stealer	Proxy	SEO fraud	Injector
Group 1	✓	✓			
Group 2	✓				
Group 3	✓				
Group 4	✓				
Group 5		✓			
Group 6		✓			
Group 7	✓				
Group 8	✓				
Group 9			✓	✓	
Group 10				✓	
Group 11	✓		✓	✓	✓
Group 12A	✓		✓	✓	✓
Group 12B	✓			✓	✓
Group 12C				✓	
Group 13	✓			✓	
Group 14				✓	✓

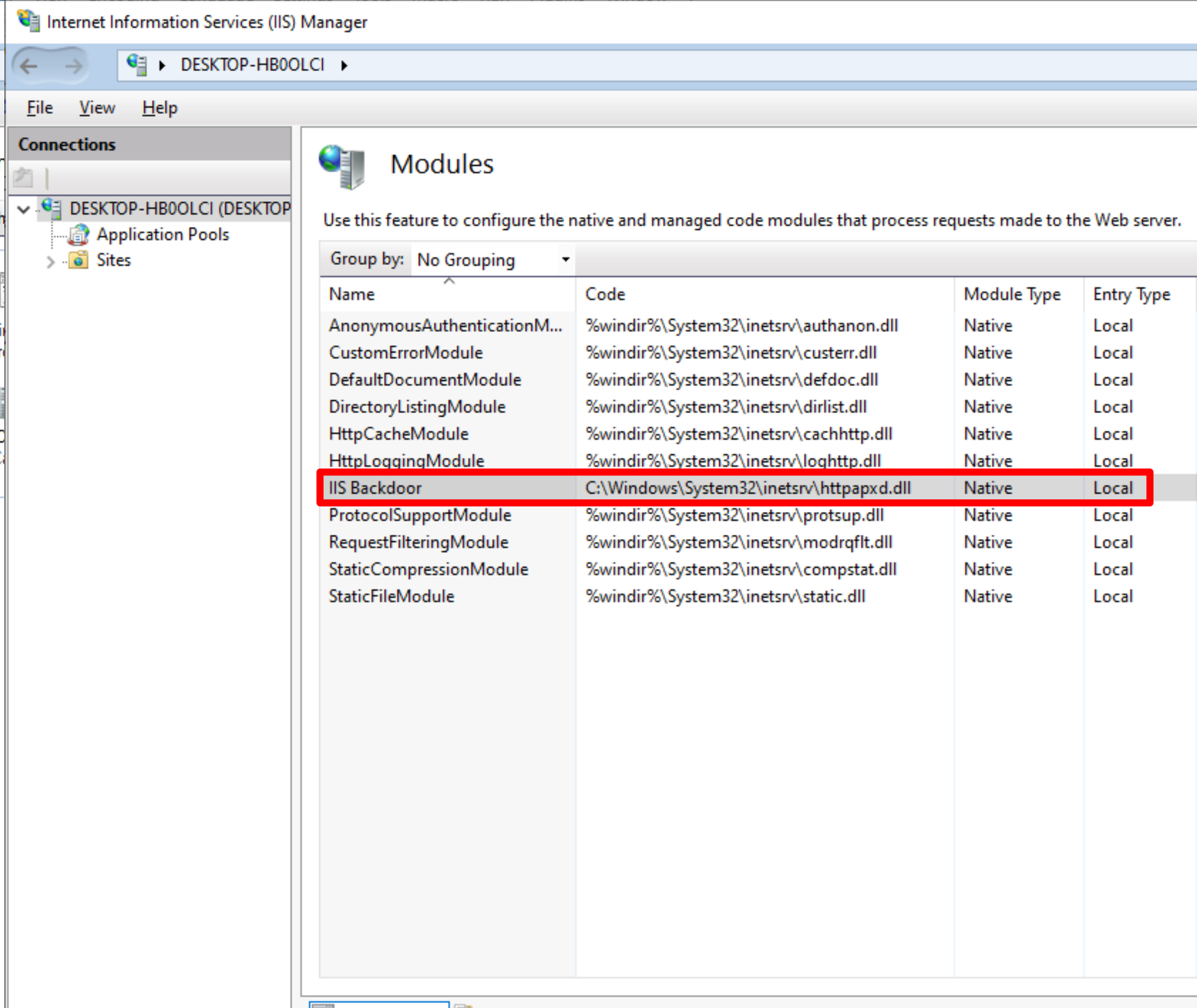
Malicious native IIS modules

Detection, mitigation and remediation

Detecting compromised servers

Inspect installed modules


Internet Information Services (IIS) Manager
DESKTOP-HB0OLCI
File View Help
Connections
DESKTOP-HB0OLCI
DESKTOP-HB0OLCI Hom
Filter:
Go
IIS
Authentic...
Compression
Default Document
Logging
MIME Types
Modules
Management
Configurat... Editor
Feature Delegation
Shared Configurat...


Internet Information Services (IIS) Manager
DESKTOP-HB0OLCI
File View Help
Connections
DESKTOP-HB0OLCI (DESKTOP
Application Pools
Sites
Modules
Use this feature to configure the native and managed code modules that process requests made to the Web server.
Group by: No Grouping
Name
Code
Module Type
Entry Type
AnonymousAuthenticationM...
%windir%\System32\inetsrv\authanon.dll
Native
Local
CustomErrorModule
%windir%\System32\inetsrv\custerr.dll
Native
Local
DefaultDocumentModule
%windir%\System32\inetsrv\defdoc.dll
Native
Local
DirectoryListingModule
%windir%\System32\inetsrv\dirlist.dll
Native
Local
HttpCacheModule
%windir%\System32\inetsrv\cachhttp.dll
Native
Local
HttpLoggingModule
%windir%\System32\inetsrv\loghttp.dll
Native
Local
IIS Backdoor
C:\Windows\System32\inetsrv\httpapxd.dll
Native
Local
ProtocolSupportModule
%windir%\System32\inetsrv\protsup.dll
Native
Local
RequestFilteringModule
%windir%\System32\inetsrv\modrqflt.dll
Native
Local
StaticCompressionModule
%windir%\System32\inetsrv\compstat.dll
Native
Local
StaticFileModule
%windir%\System32\inetsrv\static.dll
Native
Local

Detecting compromised servers

Check IIS logs

the default location is
`%SystemDrive%\inetpub\logs\LogFiles`

Inspect installed modules

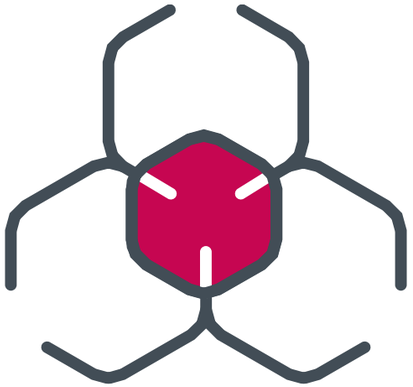
via IIS Manager, AppCmd.exe
or inspect the configuration file
`%windir%\system32\inetsrv\config\ApplicationHost.config`

Scan for known malware families

use IoCs and YARA rules listed
on our GitHub repository

DEMO

Mitigation (of compromise vectors)



Prevent server exploitation

- keep your OS up-to-date
- limit services exposed to the internet
- use strong passwords and 2FA for dedicated administrative accounts

Prevent installing malicious (e.g., trojanized) modules

- only install modules from trusted sources
- consider using an endpoint security solution

Conclusion

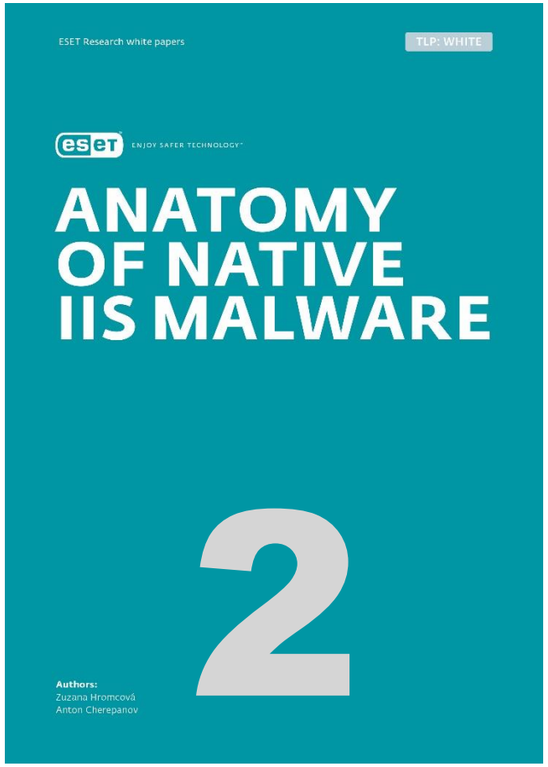
Black Hat Sound Bytes

Anatomy of Native IIS Malware

1

IIS malware: cybercrime AND cyberespionage tool

we documented 14 families (10 new); consider them in your threat model



Get the full white paper:

for a comprehensive guide on detecting, analyzing and understanding IIS malware

3

Use the IoCs and YARA rules for detection:

get them from the ESETresearch GitHub https://github.com/eset/malware-ioc/tree/master/badiis

Zuzana Hromcova
ESET Malware Researcher
@zuzana_hromcova

Thanks for watching!

www.welivesecurity.com
@ESETresearch

ESET