ESET ENJOY SAFER TECHNOLOGY

Anatomy of native IIS malware

Zuzana Hromcova | Malware Researcher

ESET ENJOY SAFER TECHNOLOGY

Anatomy of native IIS malware Internet Information Services C++ Libraries



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*





Lunt 2005 2006 ~

*Netcraft: May 2021 Web Server Survey

999 2001 2002 2004 APT 2502 FOD 2004

20%

0%

AUG 1995 1997 1998 1999 Jan Jun Nov Apr

Developer April 2021 Percent May 2021 Percent 432,167,302 35.65% 440,997,336 36.19% nginx 313,948,741 25.90% 314,774,492 25.83% Apache OpenResty 6.76% 73,839,970 6.06% 81,935,391 4.95% Microsoft 67.182.740 5.54% 60.265.118

2006 2008 2009 2011 May Oct 2 Mar 2 409

AU92 Jan 2014

Jun 2015 2016

016 2018 2019 2021 API SEP Mar 2021

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 <u>technologies overview</u> 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.





of websites use IIS software*

en joy safer technology"

Microsoft Exchange email servers with Outlook on the web



Microsoft Exchange email servers with





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):

TOTAL RESULTS 203,744 TOP COUNTRIES United States 48,776 Germany 42,877 United Kingdom 12.264 Netherlands 8.514 France 8,391 More... TOP PORTS

202.226



Modular architecture introduced

IIS v7.0

Managed modules (.NET assemblies)

2007



IIS malware reports



Menu

welivesecurity we

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





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



IIS malware reports



ESET Research white papers





ANATOMY OF NATIVE IIS MALWARE

ESET white paper +10 families

.

Aug 2021

Authors: Zuzana Hromcová Anton Cherepanov



Native IIS malware

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







Initial compromise (server exploitation, trojanized modules...) Configured as a native IIS extension (admin privileges required) Loaded by IIS Worker Process (w3wp.exe)



Can **intercept** server traffic



HTTP responses

IIS request-processing pipeline

Begin Request Processing

Authentication

Authorization

Cache Resolution

Handler Mapping

HTTP request —

Handler Pre-execution

Handler Execution

Release State

Update Cache

Update Log

End Request Processing

→ HTTP response



Server events

generate notifications



Event handlers

Server events

generate notifications



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^{to}

dg offset OnGlobalCacheCleanup dg offset OnGlobalCacheOperation dq offset OnGlobalHealthCheck dq offset OnGlobalConfigurationChange dq offset OnGlobalFileChange dg offset OnGlobalPreBeginRequest dq offset OnGlobalApplicationStart dq offset OnGlobalApplicationResolveModules dq offset OnGlobalApplicationStop dq offset OnGlobalRSCAQuery dg offset OnGlobalTraceEvent dq offset OnGlobalCustomNotification dg offset Terminate dg offset OnGlobalThreadCleanup dg offset OnGlobalApplicationPreload dg 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 CHttpModule:

; const HttpModule::`vftable'
??_7HttpModule@@6B@ dd offset OnBeginRequest

; DATA XREF: sub_7454A310+19↑o ; sub_7454A360+9↑o

Class inheriting from CGlobalModule: ; const CMyGlobalModule::`vftable' ?? 7CMyGlobalModule@@6B@ dq offset OnGlobalStopListening ; DATA XREF: DNameNode : Terminate+5^{to} dg offset OnGlobalCacheCleanup dg offset OnGlobalCacheOperation dq offset OnGlobalHealthCheck dq offset OnGlobalConfigurationChange dg offset OnGlobalFileChange dq offset OnGlobalPreBeginRequest dq offset OnGlobalApplicationStart dg offset OnGlobalApplicationResolveModules dq offset OnGlobalApplicationStop dq offset OnGlobalRSCAQuery dg offset OnGlobalTraceEvent dq offset OnGlobalCustomNotification dg offset Terminate dg offset OnGlobalThreadCleanup dg offset OnGlobalApplicationPreload dg offset OnSuspendProcess

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:

; DATA XREF: DNameNode ; Terminate+5^{to}

dg offset OnGlobalCacheCleanup dg offset OnGlobalCacheOperation dq offset OnGlobalHealthCheck dq offset OnGlobalConfigurationChange dq offset OnGlobalFileChange dg offset OnGlobalPreBeginRequest dq offset OnGlobalApplicationStart dq offset OnGlobalApplicationResolveModules dq offset OnGlobalApplicationStop dq offset OnGlobalRSCAQuery dg offset OnGlobalTraceEvent dq offset OnGlobalCustomNotification dg offset Terminate dg offset OnGlobalThreadCleanup dg offset OnGlobalApplicationPreload dg 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 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:000007FFFB1F17D6 mov ecx, 8 : Size rbx, rdx .text:000007FEFB1F17DB_mov .text:000007FEFB1F17DE call MyHttpModuleFactory_ctor rcx, ?? 7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable .text:000007FEFB1F17E3 lea .text:000007FEFB1F17EA mov cs:practory, rax .text:000007FEFB1F17F1 xor r9d, r9d .text:000007FEFB1F17F4 mov r8d, RQ SEND RESPONSE .text:000007FEFB1F17FA mov rdx, rax text:000007FFFB1F17FD_mov [rax], rcx .text:000007FEFB1F1800 mov rcx, rbx .text:000007FEFB1F1803 mov r10, [rbx] .text:000007FEFB1F1806 add rsp, 20h .text:000007FEFB1F180A pop rbx [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications] .text:000007FEFB1F180B jmp .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	<pre>lea rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable'</pre>
.text:000007FEFB1F17EA	mov cs:pFactory, rax
.text:000007FEFB1F17F1	xor 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	non chx
.text:000007FEFB1F180E	<pre>jmp [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]</pre>
.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	; Export	ted entry 1. RegisterModule
.text:000007FEFB1F17D0		
.text:000007FEFB1F17D0		
.text:000007FEFB1F17D0		
.text:000007FEFB1F17D0	public R	RegisterModule
.text:000007FEFB1F17D0	Register	rModule 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	<pre>rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory::`vftable</pre>
.text:000007FEFB1F17EA	mov	cs:practory, 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	non	chy
.text:000007FEFB1F180E	jmp	[r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]
.text:000007FEFB1F180B	Register	rModule endp
.text:000007FEFB1F180B		

1. Creates instance of the core classes

- 2. Registers module for server events
- 3. Sets priority for the module

ESCT ENJOY SAFER TECHNOLOGY

Native IIS malware Reversing



Identify implemented event handlers

📕 🚄 🖼			
text:000007FEFB1F17D0	Exported entry 1. RegisterModule		
 text:000007FEFB1F17D0			
 text:000007FEFB1F17D0			
 text:000007FEFB1F17D0			
 text:000007FEFB1F17D0	blic RegisterModule		
 text:000007FEFB1F17D0	egisterModule proc near		Kenster VIDOUNE
 text:000007FEFB1F17D0	ish rbx		
 text:000007FEFB1F17D2	ıb rsp, 20h		
 text:000007FEFB1F17D6	ov ecx, 8 ; Size		
 text:000007FEFB1F17DB	ov rbx, rdx		
 text:000007FEFB1F17DE	11 MyHttpModuleFactory_ctor		
 text:000007FEFB1F17E3	<pre>a rcx, ??_7CMyHttpModuleFactory@@6B@ ; const CMyHttpModuleFactory</pre>	<pre>/::`vftable'</pre>	
 text:000007FEFB1F17EA	ov cs:pFactory, rax		
 text:000007FEFB1F17F1 :	or r9d, r9d		
 text:000007FEFB1F17F4	v r8d, RQ_SEND_RESPONSE		
 text:000007FEFB1F17FA ı	ov rdx, rax		
 text:000007FEFB1F17FD	ov [rax], rcx		
 text:000007FEFB1F1800	ov ncx, nbx		
 text:000007FEFB1F1803	ov r10, [rbx]		
 text:000007FEFB1F1806	ld rsp, 20h		
 text:000007FEFB1F180A	op rbx		
 text:000007FEFB1F180B	<pre>p [r10+IHttpModuleRegistrationInfoVtbl.SetRequestNotifications]</pre>		
 text:000007FEFB1F180B	egisterModule endp		
 text:000007FEFB1F180B			

Identify implemented event handlers

```
; const HttpModule::`vftable'
```

??_7HttpModule@@6B@ dd offset OnBeginRequest

dd offset sub 74549CD0

; DATA XREF: sub_7454A310+19↑o

; sub_7454A360+91o

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 offeat sub 7454A1E0

.text:00000000001417A0	
.text:0000000001417A0	
.text:0000000001417A0	
.text:0000000001417A0	OnSendResponse proc near
.text:0000000001417A0	sub rsp, 28h
.text:0000000001417A4	<pre>lea rcx, OutputString ; "This module subscribed to event "</pre>
.text:0000000001417AB	call cs:OutputDebugStringA
.text:0000000001417B1	<pre>lea rcx, aChttpmoduleOns ; "CHttpModule::OnSendResponse"</pre>
.text:0000000001417B8	call cs:OutputDebugStringA
.text:0000000001417BE	<pre>lea rcx, aButDidNotOverr ; " but did not override the method in it</pre>
.text:0000000001417C5	call cs:OutputDebugStringA
.text:0000000001417CB	call cs:DebugBreak
.text:0000000001417D1	xor eax, eax
.text:00000000001417D3	add rsp, 28h
.text:00000000001417D7	retn
.text:0000000001417D7	OnSendResponse endp
.text:0000000001417D7	
L	-71

default method



Identify implemented event handlers

eset

```
; const HttpModule::`vftable'
                                                    malicious handlers
  ?? 7HttpModule@@6B@ dd offset OnBeginRequest
                                             ; DATA XREF: sub 7454A310+1910
                                             ; sub 7454A360+91o
                   dd offset sub 74549CD0
                   dd offset sub 74549D00
                   dd offset sub 74549D30
                                                        1 int thiscall OnLogRequest(httpModuleObj *this, int pHttpContext, int pProvider)
                   dd offset sub 74549D60
                                                        2 {
                   dd offset sub 74549D90
                                                            int pHttpRequest; // edi
                                                        3
                   dd offset sub 74549DC0
                                                            int pHttpResponse; // eax
                                                        4
                   dd offset sub 74549DF0
                                                        5
                   dd offset sub 74549E20
                                                        6
                                                            pHttpRequest = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetRequest)))(pHttpContext);
                   dd offset sub 74549E50
                                                            pHttpResponse = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetResponse)))(pHttpContext);
                                                        7
                                                        8
                                                            if ( pHttpRequest
                   dd offset sub 74549E80
                                                        9
                                                              && pHttpResponse
                   dd offset sub 74549EB0
                                                              && ((this->flagIgnoreRequest & 1) != offsetof(httpModuleObj, field 0) || this->flagAttackerRequest) )
                                                       10
                   dd offset sub 74549EE0
                                                       11
                   dd offset sub 74549F10
                                                       12
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetHttpMethod)))(pHttpRequest, "GET");
                   dd offset sub 74549F40
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetUrl)))(pHttpRequest, "/", 1, 1);
                                                       13
                   dd offset sub 74549F70
                                                       14
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Cookie");
                   dd offset sub 74549FA0
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Origin");
                                                       15
                   dd offset sub 74549FD0
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Referer");
                   dd offset sub 7454A000
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Mode");
                                                       18
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Sec-Fetch-Site");
                   dd offset sub 7454A030
                                                       19
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Type");
                   dd offset OnLogRequest
                                                       20
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Length");
                   00 OTTSET SUD /4548090
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-IP");
                                                       21
                   dd offset OnEndRequest
                                                       22
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-For");
                   dd offset sub 7454A0F0
                                                       23
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-By");
                   dd offset sub 7454A120
                                                       24
                                                              (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "X-Forwarded-Proto");
                   dd offset sub 7454A150
                                                       25
ENIC
                   dd offset sub 7454A180
                                                       26
                                                            return 0;
                                                      27 }
                   dd offset sub 7454A1B0
                   dd offeat sub 7454A1E0
```

- 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

1 int thiscall OnLogRequest(httpModuleObj *this, int pHttpContext, int pProvider) 2 int pHttpRequest; // edi int pHttpResponse; // eax Request = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetRequest)))(pHttpContext) HttpResponse = (*(*pHttpContext + offsetof(IHttpContext2Vtbl, GetResponse)))(pHttpContex 9 && pHttpResponse && ((this->flagIgnoreRequest & 1) != offsetof(httpModuleObj, field 0) || this->flagAttackerRequest)) 10 11 12 *(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetHttpMethod)))(pHttpRequest, "GET"); *(*pHttpRequest + offsetof(IHttpRequest2Vtbl, SetUrl)))(pHttpRequest, "/", 1, 1); 13 *(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Cookie"); 14 15 (*(*pHttpRequest + offsetof(1HttpRequest2Vtb1, DeleteHeader)))(pHttpRequest, 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"); (*(*pHttpRequest + offsetof(IHttpRequest2Vtbl, DeleteHeader)))(pHttpRequest, "Content-Length"); 20 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



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





ESET ENJOY SAFER TECHNOLOGY

Known IIS malware families

see our paper for detailed analyses

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



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)

EN IOY SAFER TECHNOLOGY"

eset



Targets e-commerce websites (IIS infostealers)



Aids in C&C routing (IIS proxies)



Manipulates SERP (SEO fraud)



Aids in malware distribution (IIS injectors)

welivesecurity

welivesecurity weser

IIStealer: A server-side threat to e-commerce transactions

The first in our series on IIS threats looks at a malicious IIS extension that intercepts server transactions to steal credit card information

Zuzana Hromcová

6 Aug 2021 - 03:00PM

IISpy: A complex server-side backdoor with anti-forensic features

The second in our series on IIS threats dissects a malicious IIS extension that employs nifty tricks in an attempt to secure long-term espionage on the compromised servers



9 Aug 2021 - 11:30AM

welivesecurity

IlSerpent: Malware-driven SEO fraud as a service

The last in our series on IIS threats introduces a malicious IIS extension used to manipulate page rankings for third-party websites

Zuzana Hromcová

11 Aug 2021 - 11:30AM

Read our blogpost series IIS threats under the microscope

Read the white paper

in the conference proceedings



7 - 8 October, 2021 / vblocalhost.com

ANATOMY OF NATIVE IIS MALWARE,

Zuzana Hromcová ESET, Slovakia

zuzana.hromcova@eset.com







Zuzana Hromcova

ESET Malware Researcher @zuzana_hromcova

Anton Cherepanov

ESET Senior Malware Researcher

@cherepanov74





THANK YOU!

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