Joan Calvet
@joancalvet

Marion Marschalek
@pinkflawd

Paul Rascagnères
@r00tsbsd

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

SNOWGLOBE:

From Discovery to Attribution

[redacted]
CSEC CNT / Cyber CI
SIGDEV 2011 Cyber Thread

Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

Canada

Once upon a time...

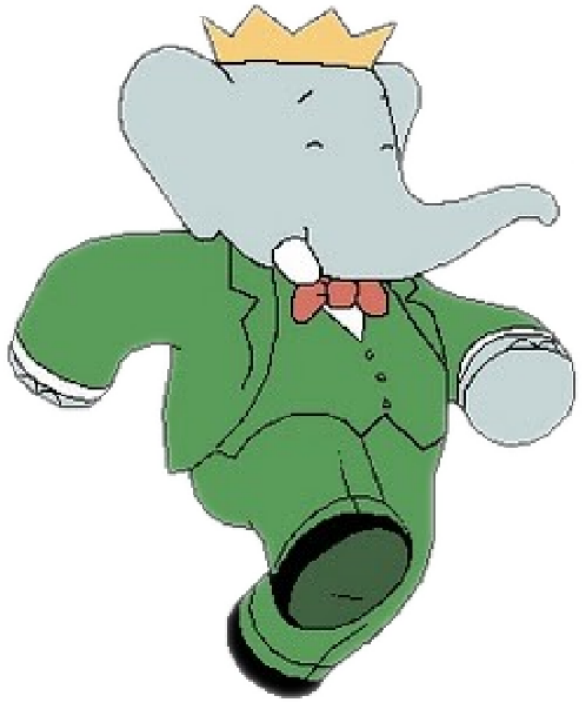
SNOWGLOBE.

- CSEC assesses, with moderate certainty, SNOWGLOBE to be a state-sponsored CNO effort, put forth by a French intelligence agency

Once upon a time...

Attribution: Binary Artifacts

- ntrass.exe
  - DLL Loader uploaded to a victim as part of tasking seen in collection
  - Internal Name: Babar
  - Developer username: titi
- Babar is a popular French children's television show
- Titi is a French diminutive for Thiery, or a colloquial term for a small person



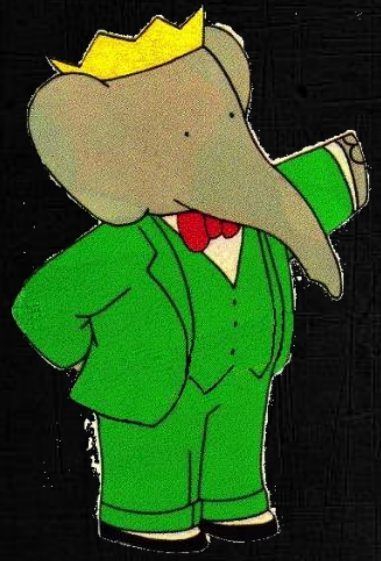
Once upon a time...


LET THE HUNT BEGIN!

TFC

NGBD

NBOT



2009

2011

2014

TIME

NBOT

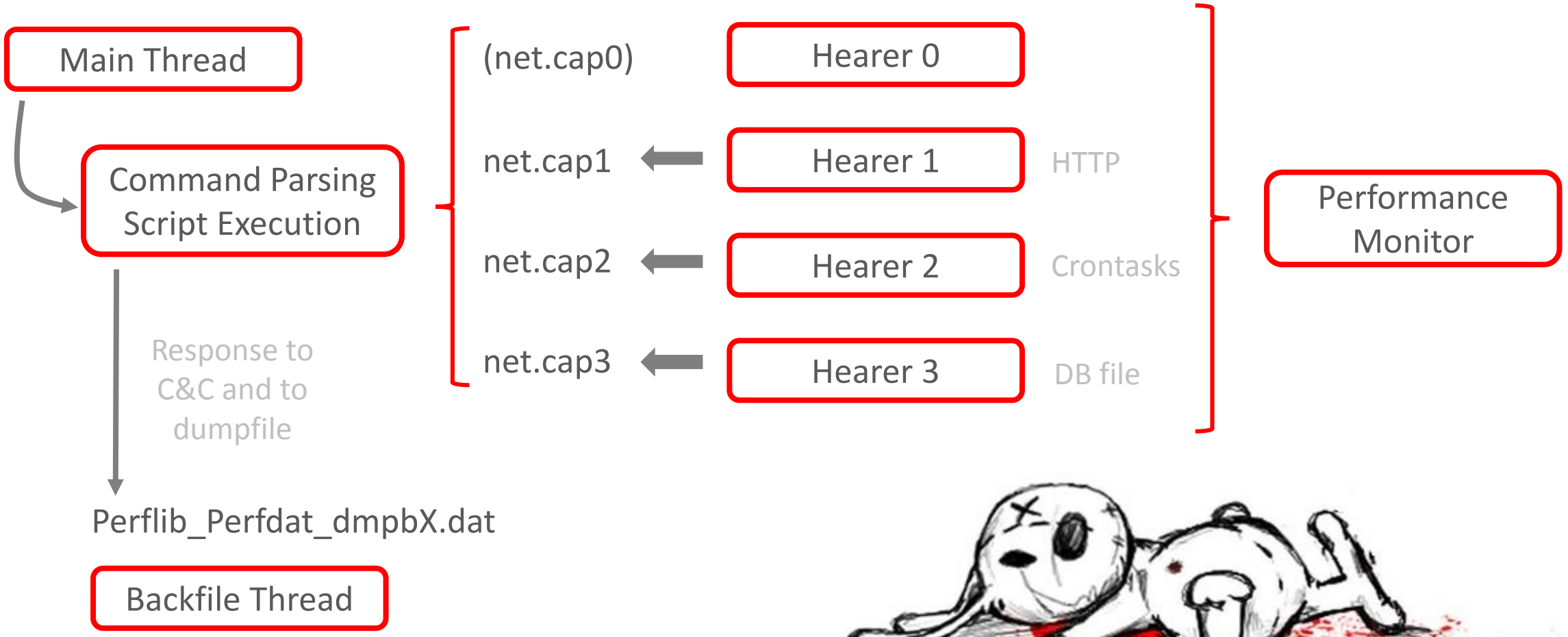
```
unicode 0, <HTTPF>,0
                        ; DATA XREF: cto
                        ; ctor_ASPFLOOD+
unicode 0, <ASPFLOOD>,0
db    0
db    0
                        ; DATA XREF: cto
                        ; ctor_TCPFLOOD+
unicode 0, <TCPFLOOD>,0
db    0
db    0
                        ; DATA XREF: cto
                        ; ctor_WEBFLOOD+
unicode 0, <WEBFLOOD>,0
db    0
db    0
                        ; DATA XREF: cto
                        ; ctor_POSTFLOOD
unicode 0, <POSTFLOOD>,0
                        ; DATA XREF: cto
                        ; ctor_STATISTIC
unicode 0, <STATISTICS>,0
db    0
db    0
                        ; DATA XREF: cto
                        ; ctor_KILL+18↑o
unicode 0, <KILL>,0
db    0
```

Obviously DDoS

No packer or crypter

C&Cs sinkholed by Kaspersky

SCRIPTABLE BOT through lua script injection



RABBIT ARMOURING

Emulator check

Containing directory name check

Payload‘s creation time stamp changed

Number of running processes 15+

Time API hook detection

Obfuscation of subset of APIs

Infection ‚strategy‘

Payload only started on reboot

BABAR

M International

Quand les Canadiens partent en chasse de « Babar »

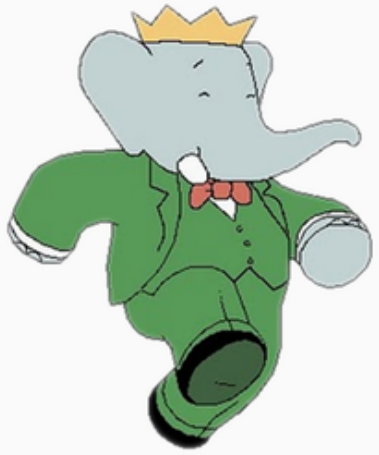
Le Monde | 21.03.2014 à 12h26 • Mis à jour le 19.05.2014 à 14h13 |

Par Jacques Follorou et Martin Untersinger

ntrass.exe
- DLL Loader uploaded to a victim as part of tasking seen in collection
- Internal Name: Babar
- Developer username: titi

Babar is a popular French children's television show

Titi is a French diminutive for Thiery, or a colloquial term for a small person



C'est une véritable traque qu'ont menée les services secrets techniques canadiens du Centre de la sécurité des télécommunications du Canada (CSEC). Elle est relatée dans le document fourni au *Monde* par Edward Snowden, dans lequel ils présentent leurs trouvailles. Avare en détails, ce document permet néanmoins de retracer l'enquête qui a permis de pointer la France du doigt.

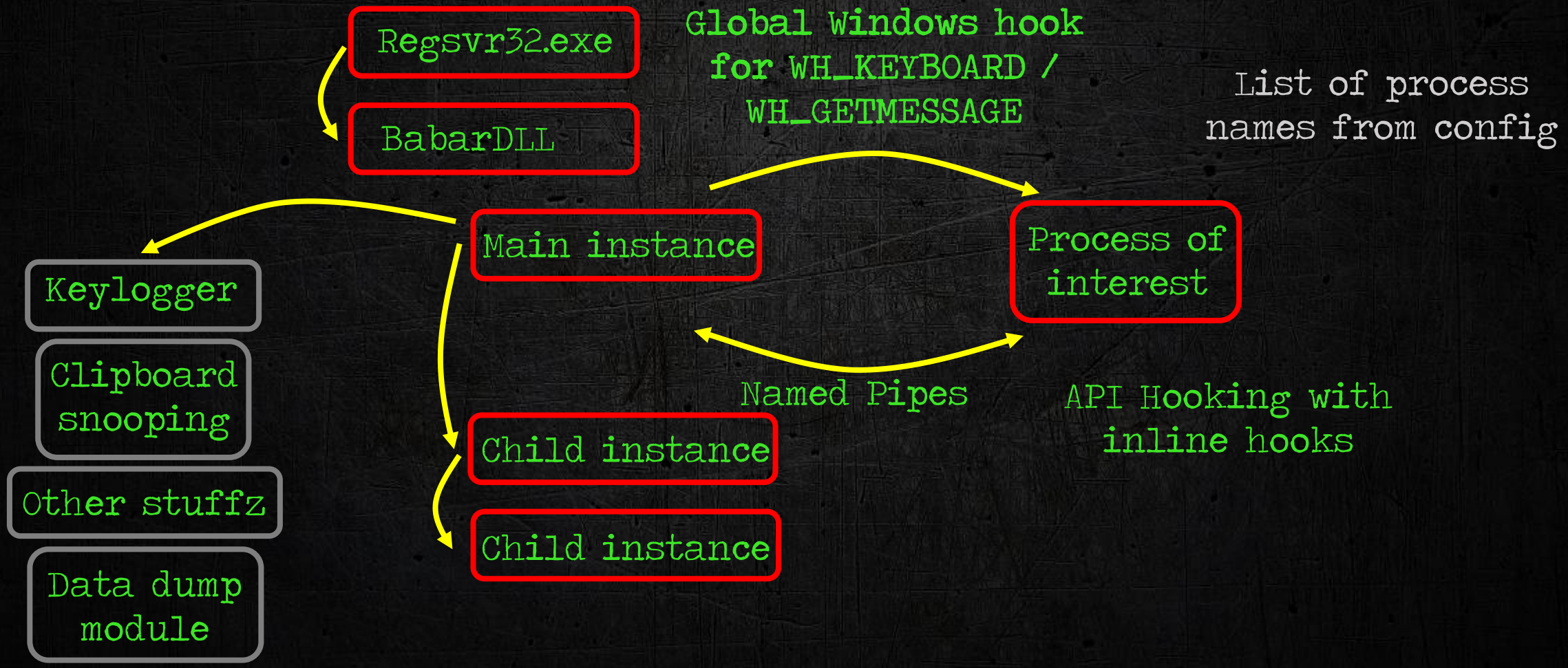
Comme dans une partie de chasse, ce sont des empreintes qui attirent en premier lieu l'attention des services canadiens. La note interne indique en effet que le CSEC collecte quotidiennement et automatiquement un certain nombre de

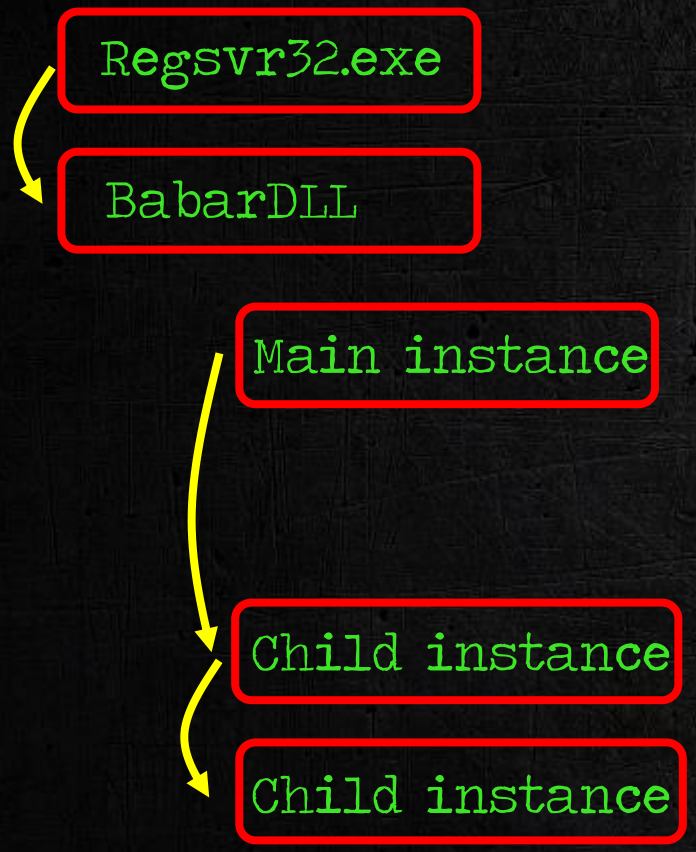
Babar

PET Persistent Elephant Threat

- Espionage par excellence
  - Keylogging, screenshots, audio captures, clipboard data, what-not.
- Via local instance or through:
  - hooking APIs in remote processes
    - after invading them via global Windows hooks

Modus Operandi Elephanti





Create section object with crucial information
- Pipe name
- number of existing instances
- export name to be called

Copy function stub to target process memory

Create remote thread
- loads Babar DLL
- calls indicated export
- Hands over data from shared object

Happily run DLL

Invisible message-only window

Message dispatching

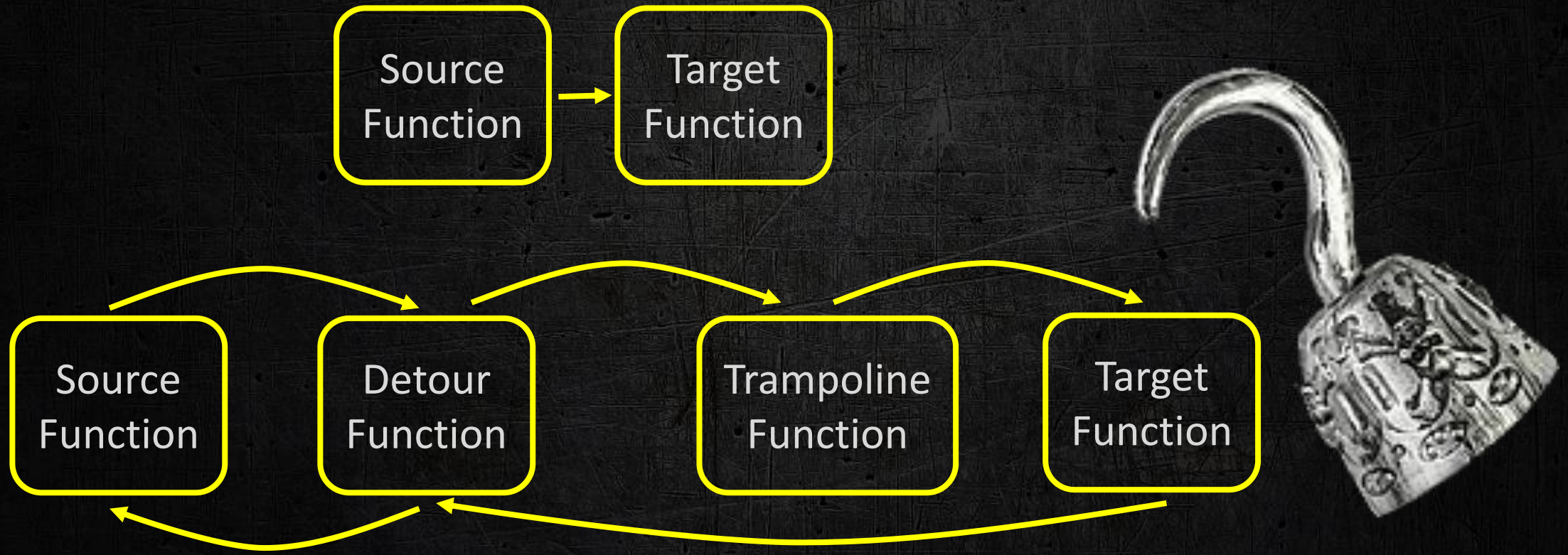
- Receive WM_INPUT register raw input device with RAWINPUTDEVICE struct as follows:
  - Set RIDEV_INPUTSINK flag – receive system wide input
  - usUsagePage set to 1 – generic desktop controls
  - usUsage set to 6 – keyboard

On WM_INPUT call GetRawInputData

Map virtual key code to character & log to file

Hiding in plain sight

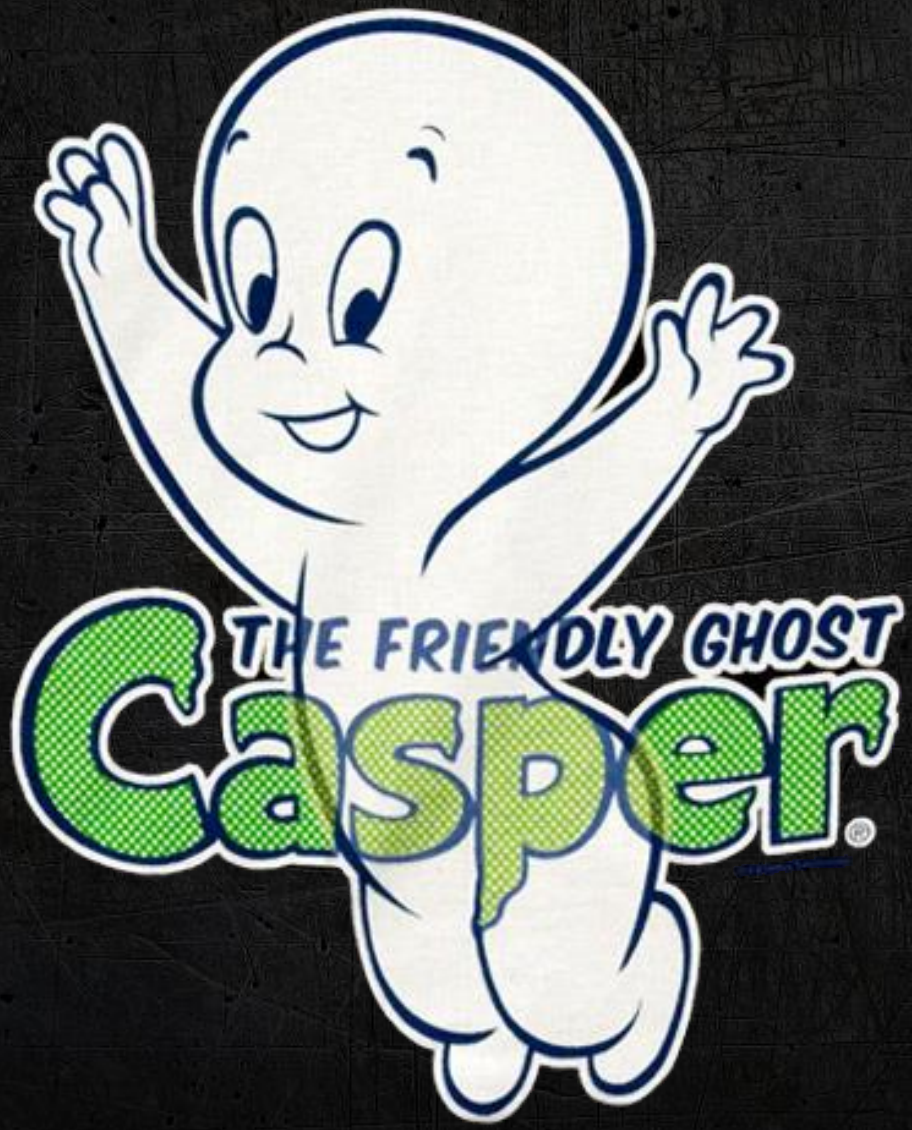
Rooootkittykittykitty



Internet communication | File creation | Audio streams

http://research.microsoft.com/en-us/projects/detours/

“To people who ask me to compare the complexity of #Regin and #Babar, keep in mind that a Peugeot is enough for the day-to-day life ;)” – Paul Rascagnères


THE FRIENDLY GHOST
Casper

Casper is a reconnaissance tool developed in C++

```
;
; Export Address Table for Casper_DLL.dll
;
```

Deployed in April 2014 on Syrian targets through a Flash 0day (CVE-2014-0515)

Exploit + Casper binaries + C&C server all hosted on website of Syrian Justice Ministry

Casper Playing Chess Against AVs

```
<STRATEGY RUNKEY="API" AUTODEL="DEL" INJECTION="YES" SAFENOTIF="YES" SERVICE="NONE" ESCAPE="NO">

    <AV NAME="BitDefender Antivirus"
        RUNKEY="API"
        AUTODEL="API"
        INJECTION="NO"
        SAFENOTIF="YES"/>

    <AV NAME="avast! Antivirus"
        RUNKEY="WMI"
        AUTODEL="WMI"
        INJECTION="NO"
        SAFENOTIF="YES"
        ESCAPE="YES"/>

[...REDACTED...]

</STRATEGY>
```

```
BOOL __stdcall Autodel_API::DeleteFile(LPCWSTR file_to_delete)
{
  return MoveFileExW(file_to_delete, 0, MOVEFILE_DELAY_UNTIL_REBOOT);
}
```

```
int __thiscall Autodel_WMI::DeleteFile(void *this, int a2)
{
  v2 = this;
  v3 = (wchar_t *)DecryptString(&unk_417688);
  // "cmd.exe /C FOR /L %%i IN (1,1,%d) DO
  //   IF EXIST "%ws" (DEL "%ws" & SYSTEMINFO)
  //   ELSE EXIT"
  FormatStr(&Dst, v3, 0);
  return WMI::ExecuteProcess((int)v2 + 4, (int)&Dst);
}
```

Payload Installation

```
int __thiscall DLL::GetProcAddressFromHash(dll_resol *this, int arg_hash)
{
  // hash_to_look_for should be equal to arg_hash
  hash_to_look_for = 12345678 ^ this->checksum ^ arg_hash;
  ...
```

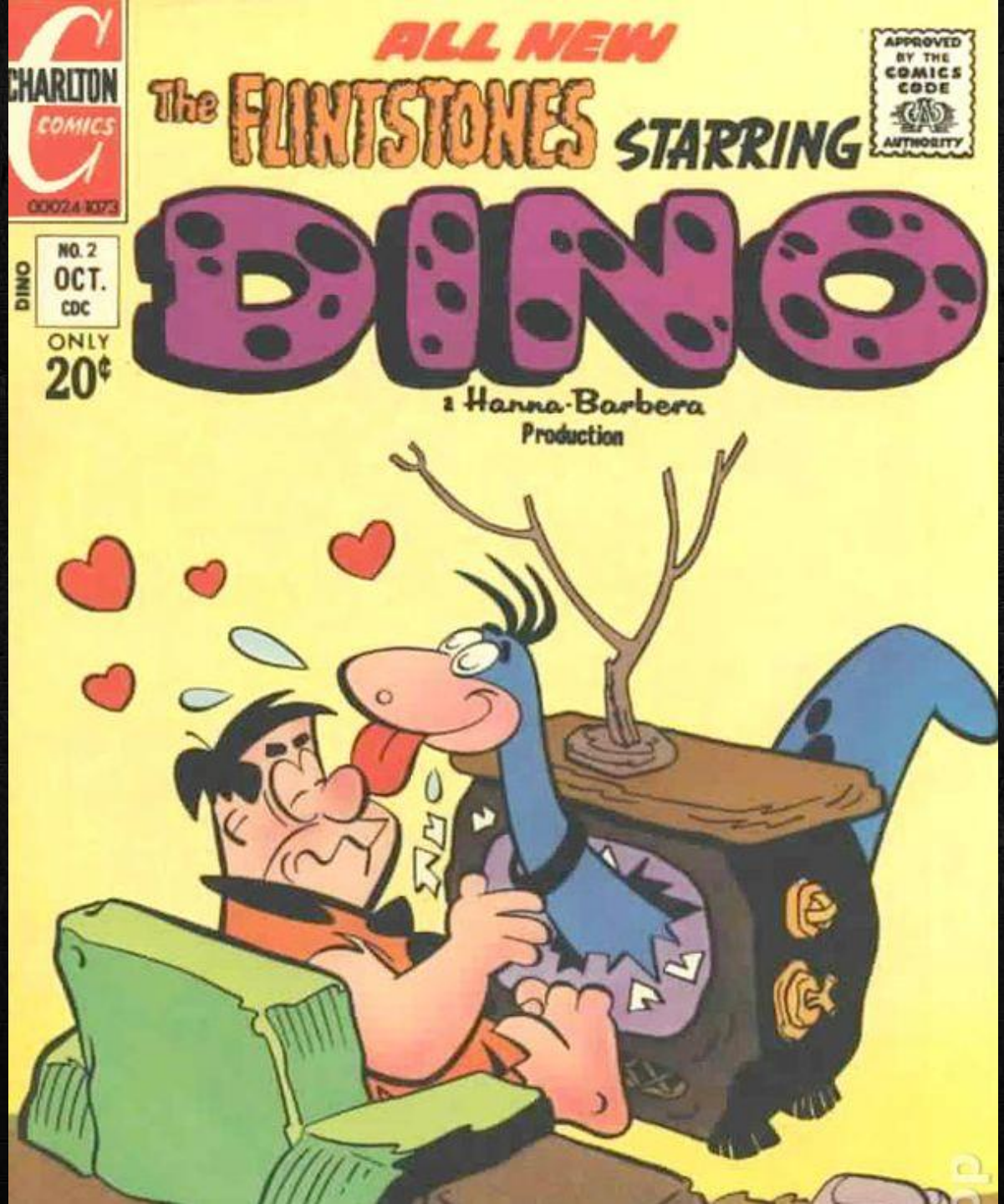
```
LOBYTE(argv[1]) = BYTE3(argv[1]) ^ argv[1];
BYTE1(argv[1]) ^= BYTE3(argv[1]);
HIWORD(argv[1]) = (BYTE3(argv[1]) ^ BYTE2(argv[1]));
DLL_object->checksum = (int)argv[1];
```

Crash when Casper calls the (wrong) retrieved address!

Detailed report sent to C&C

C&C sends back XML file indicating payload to deploy

```
***** SECURITY INFORMATION *****
AntiVirus: N/A
Firewall: N/A

***** EXECUTION CONTEXT *****
Version: 4.4.1
...[REDACTED]...

***** SYSTEM INFORMATION *****
Architecture: x86
OS Version: 5.1
Service Pack: Service Pack 3
Default Browser: firefox.exe
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Win32)
Organization:
Owner: john
Country: United States

***** Running PROCESS *****
...[REDACTED]...

*****HKLM AutoRun x86 PROCESS *****
...[REDACTED]...

*****HKLM AutoRun x64 PROCESS *****
...[REDACTED]...
```


ALL NEW
The FLINTSTONES STARRING
DINO
a Hanna-Barbera Production
CHARLTON COMICS
APPROVED BY THE COMICS CODE AUTHORITY
DINO
NO. 2
OCT.
CDC
ONLY
20¢

Espionage backdoor with numerous features

For example, complex file search requests:

"Give me all files with .doc extension, whose size is greater than X bytes and were modified in the last Y days"

Popped up in Iran in 2013

Developed in C++ in a modular fashion

No RTTI, but many verbose error messages

"Date is invalid ! Date Format is ddmmyyyy"

"decyphering failed on bd"

"Can't change the past, sorry..."

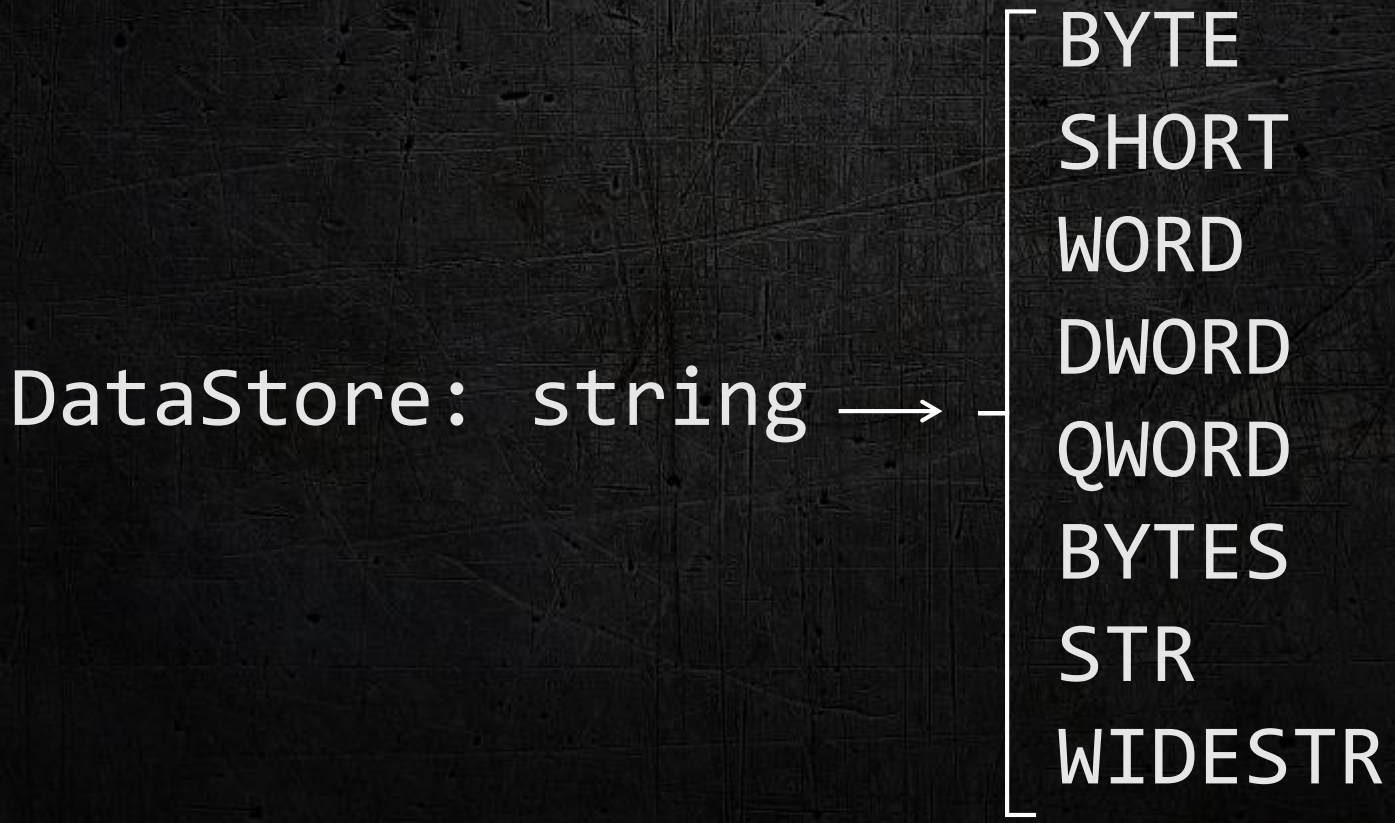

```
;
; Export Address Table for Dino.exe
;
```

Dino Modules

Module Name	Purpose
PSM	Encrypted on disk copy of Dino modules
CORE	Configuration storage
CRONTAB	Tasks scheduler
FMGR	Files upload and download manager
CMDEXEC	Commands execution manager
CMDEXECQ	Storage queue for commands to execute
ENVVAR	Storage for environment variables

DataStore

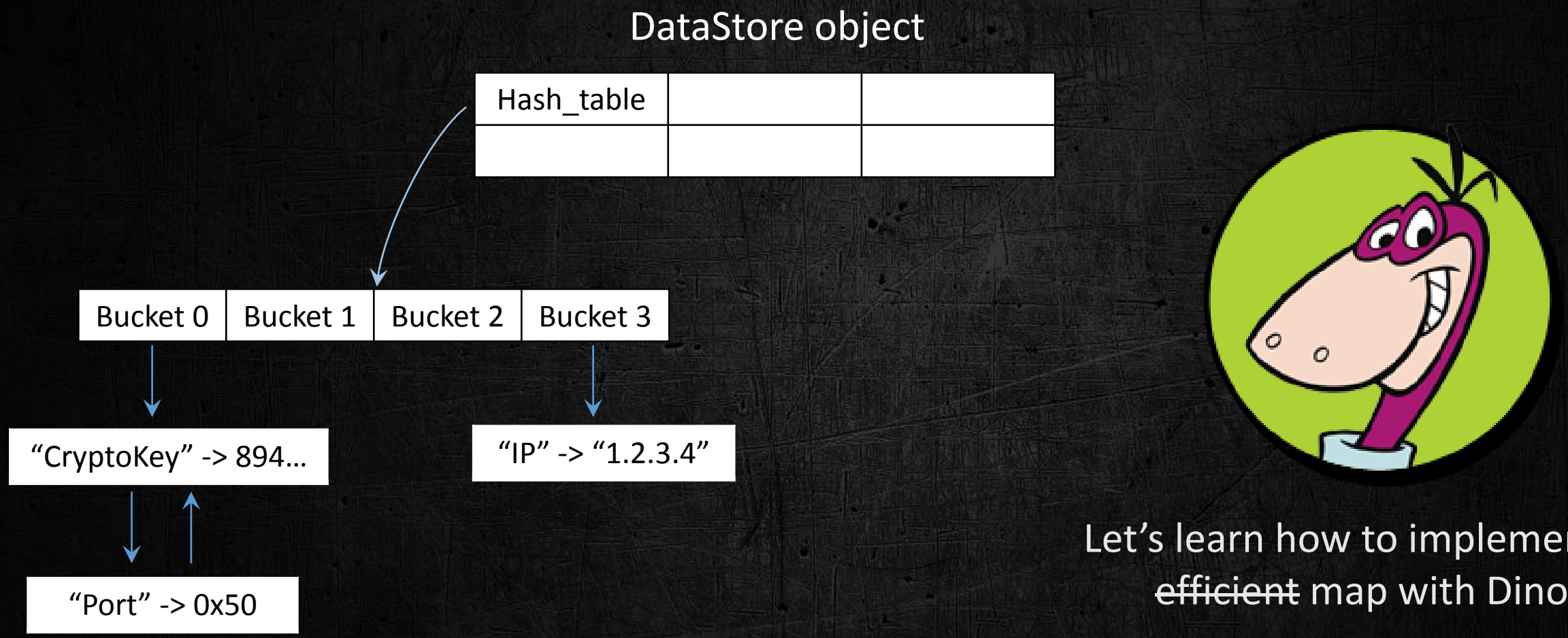
Custom data structure used *a lot* in Dino



Example: CORE module DataStore

(Dino's Config)

```
recID:      11173-01-PRS                          WIDESTR
Version:    1.2                                   WIDESTR
BD_Keys:    4D414…[REDACTED]…B3506                BYTES
MaxDelay:   00000E10                              DWORD
ComServer0: hXXp://azhar.bf/[REDACTED].php        STR
…
```

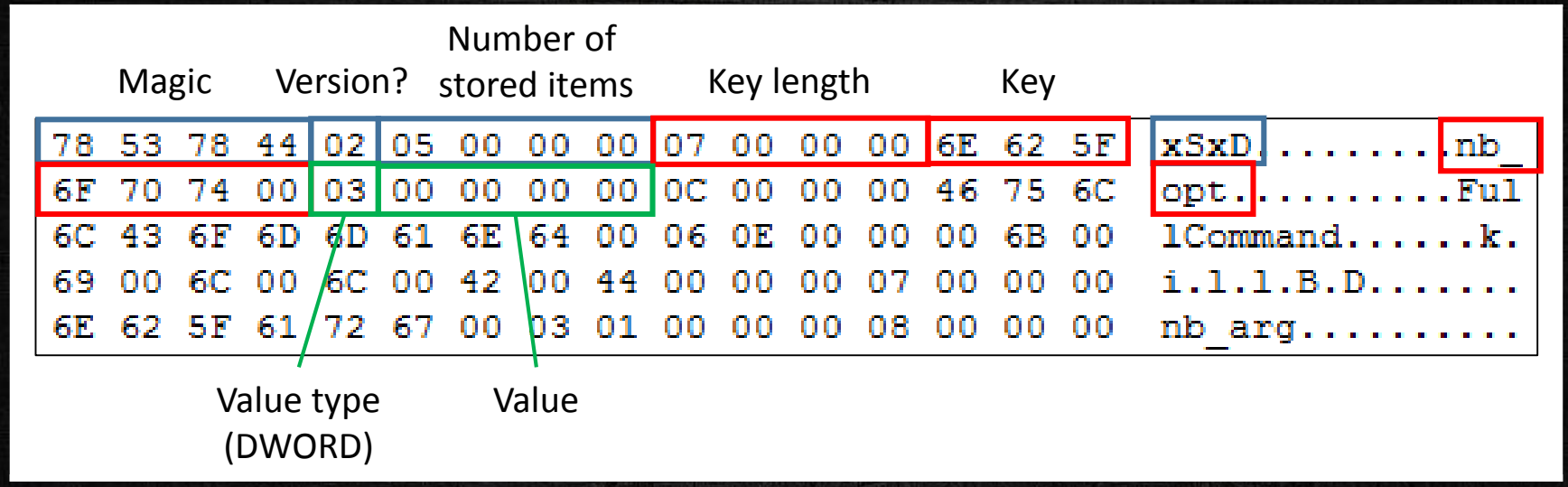


Let’s learn how to implement an ~~efficient~~ map with Dino!

Hash(“IP”) = 3 mod 4

Hash(“CryptoKey”) = Hash(“Port”) = 0 mod 4

DataStore Serialization



Example of usage: PSM module saves Dino modules as serialized DataStore into an encrypted file

```
v1->rc4_key = "PsmIsANiceM0du1eWith0SugarInsideA";
v1->len_rc4_key = 32;
```

RamFS

Temporary “file-system” mounted in memory from an encrypted blob stored in Dino configuration

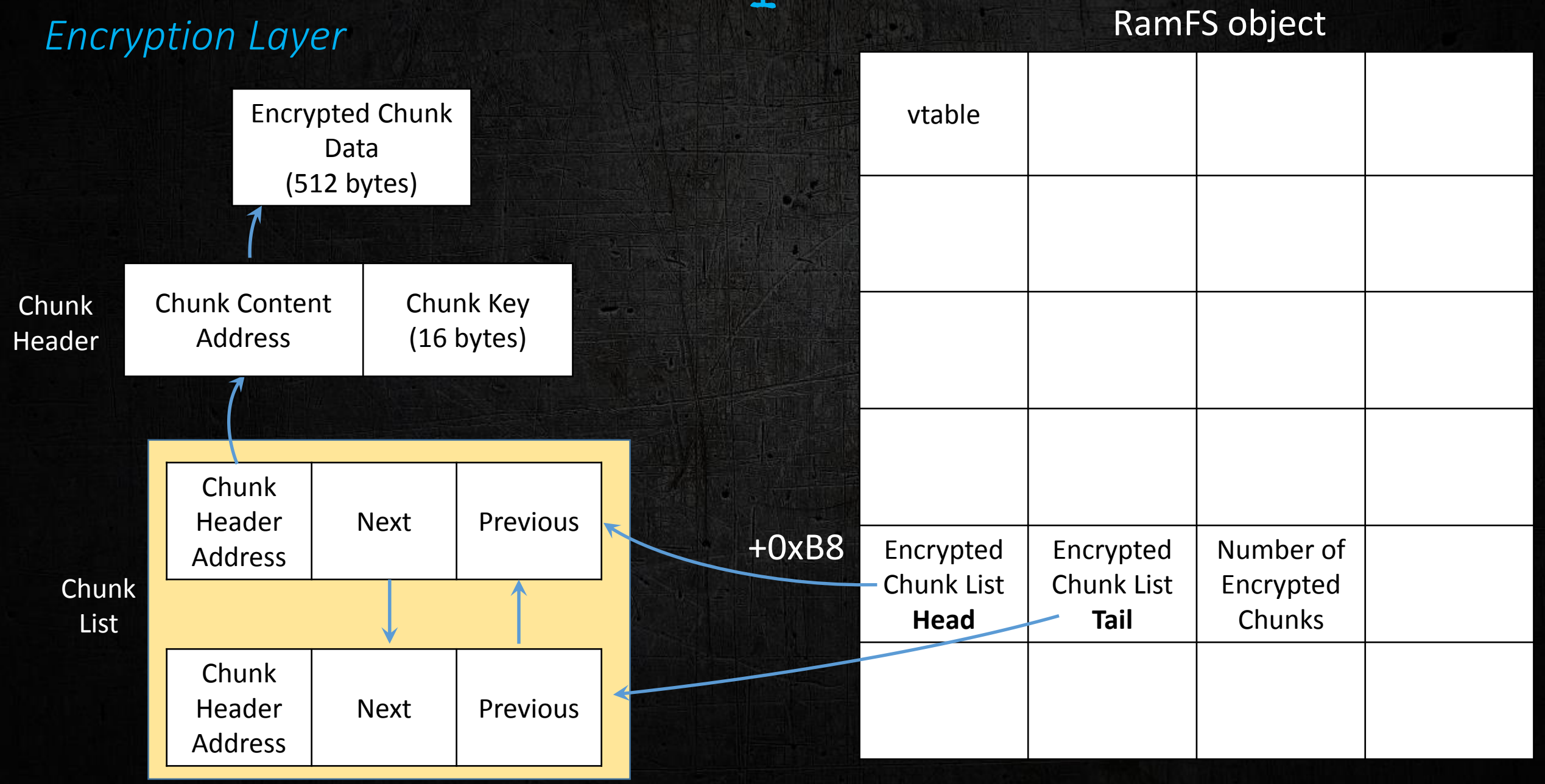
Once mounted, RamFS remains stored in encrypted chunks, decrypted on-demand

In our Dino sample, RamFS initially contains a “cleaner” file, which is executed to remove the malware from the system

```
cleaner_file_name = DataStore::SearchForKey(dino_config_datastore, k_cleaner_file_name);
if ( cleaner_file_name && cleaner_file_name->type == WIDESTR )
{
  ramfs_crypto_key = DataStore::SearchForKey(dino_config_datastore, k_ramfs_crypto_key);
  if ( ramfs_crypto_key && ramfs_crypto_key->type == STR )
  {
    if ( MountRamFS(...)
    {
      ExecuteCleanerRamFS((int)&var_ramfs_obj);
      DataStore::StoreValue(v14, "results", L"cleaner executed, exiting", a1);
      DataStore::StoreValueFixedSize(arg_datastore, "destroyPSM", 1, 0, 1);
    }
    else
    {
      DataStore::StoreValue(v11, "results", L"Unable to mount cleaner RamFS, exiting", a1);
    }
  }
  else
  {
    DataStore::StoreValue(v10, "results", L"No cleaner Passphrase Found, exiting", a1);
  }
}
else
{
  DataStore::StoreValue(a1, "results", L"No cleaner Script Found, exiting", a1);
}
```

RamFS Low-Level Implementation

Encryption Layer



RamFS Low-Level Implementation

File system

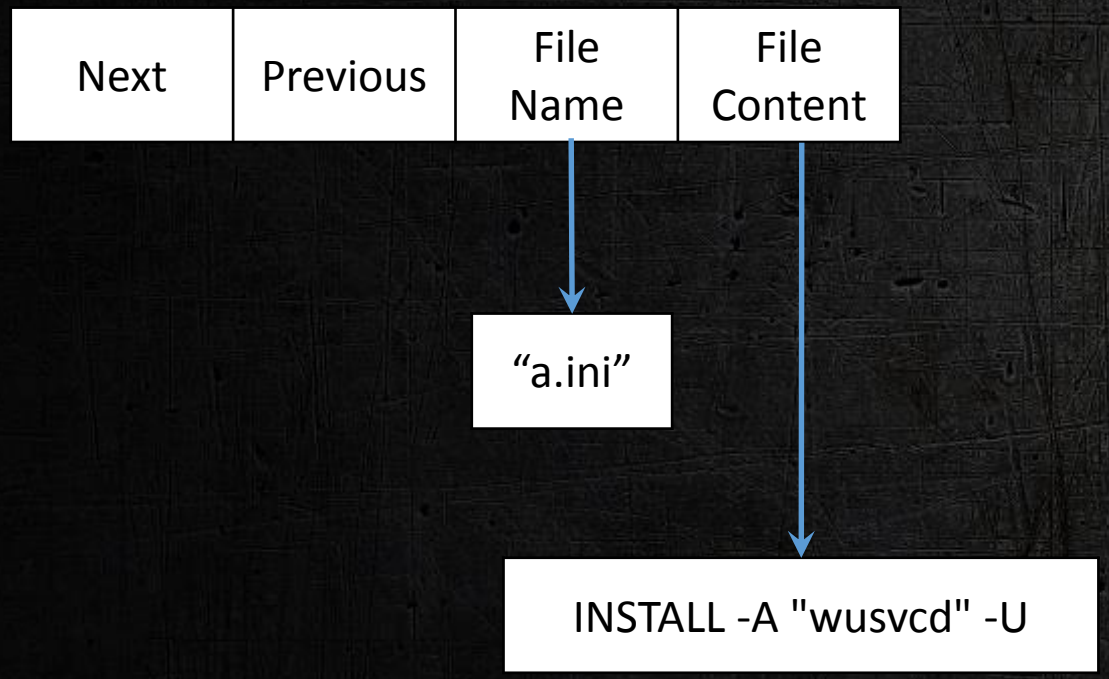
- The file system structure is decrypted from the first chunk and stored at the beginning of RamFS object
- The rest of the file system will be decrypted on-demand, and then re-encrypted
- The name of three fields is given in error messages displayed when a checksum verification fails

RamFS object

vtable			
	“File_List”	“FreeChunk Block_List”	“FreeFile Header_List”
Encrypted Chunk List **Head**	Encrypted Chunk List **Tail**	Number of Encrypted Chunks	

RamFS Low-Level Implementation

File system



RamFS object

vtable			
	“File_List”	“FreeChunk Block_List”	“FreeFile Header_List”
Encrypted Chunk List **Head**	Encrypted Chunk List **Tail**	Number of Encrypted Chunks	

RamFS Commands

Command	Purpose
INSTALL	Triggers installation or uninstallation of the malware
EXTRACT	Extracts a file stored in RamFS to the real file system
EXEC	Executes a file stored in RamFS
INJECT	Injects a file stored in RamFS in a designated process
KILL	Terminates a running process

Is RamFS Custom?

File names and file content are in Unicode

Maximum file name length is 260 characters

Unencrypted chunks are 540 bytes length

No metadata on files (?)

The link©?

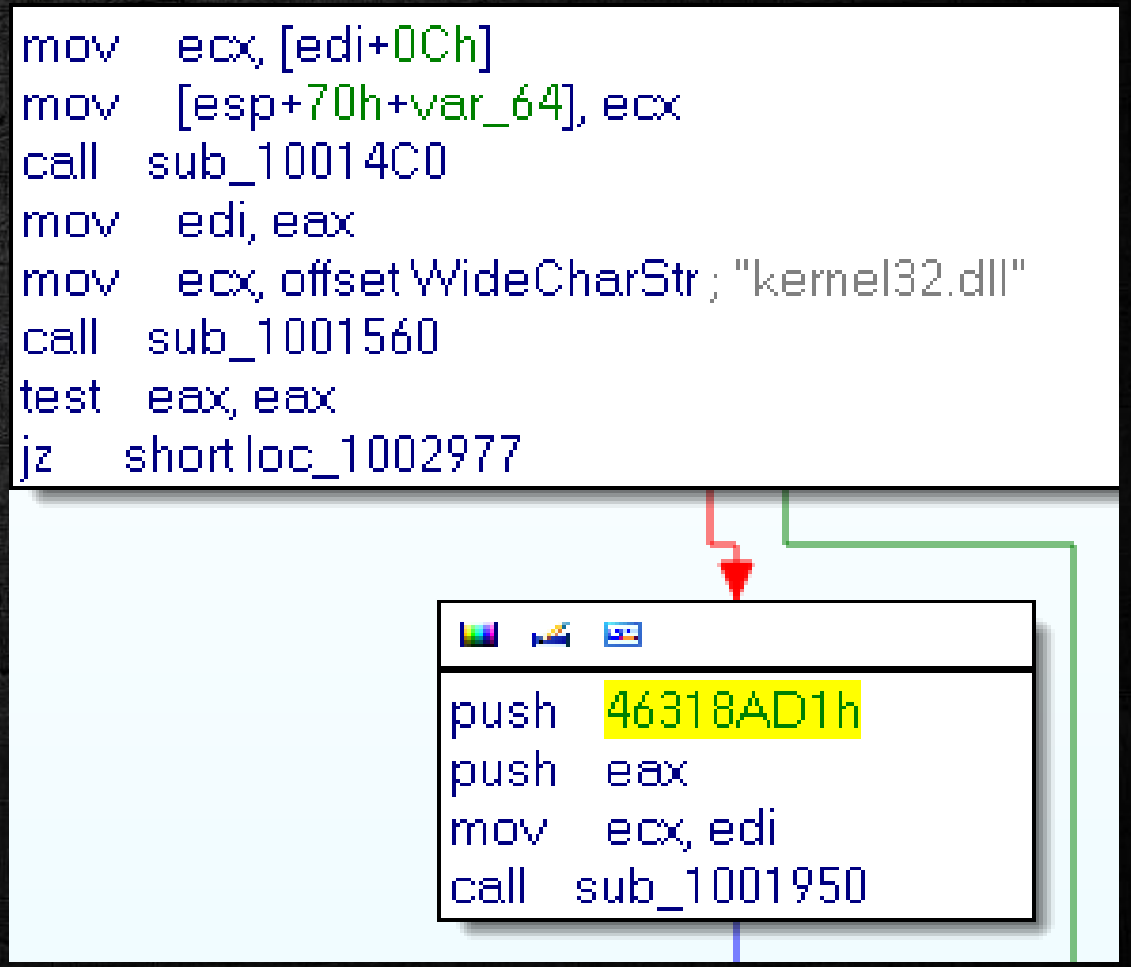

COSPLAY
I'd be upset too if i had to save that ugly thing.
\o/ MotivatedPhotos.com

API obfuscation

All Cartoons use the same approach:

1. Load the library in memory
2. Generate a hash for each exported function name
3. Check if the generated hash is equal to the hash that the malware wants to execute
4. if yes, execute the function

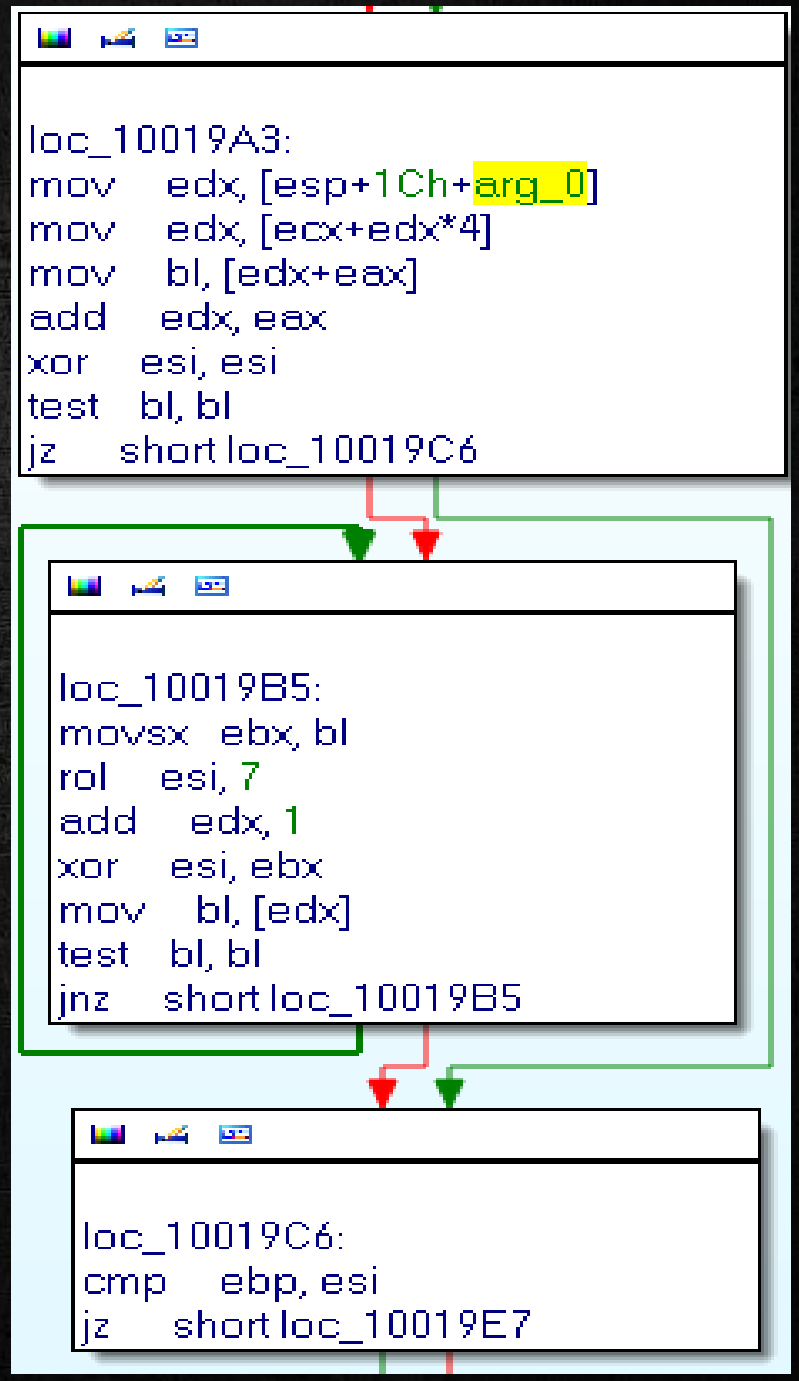
We identified 2 hash algorithms



API obfuscation

Algorithm used by Bunny & Casper

```
#!/usr/bin/python
CRC = 0
function = "CreateProcessW"
for i in list(function)
  key = rol32(CRC, 7)
  CRC = ord(i)^key
print function+": 0x%08x" % (CRC)

CreateProcessW: 0x46318ad1
```



AV identification

WMI query

Windows Security Center WMI providers:

- ROOT\SecurityCenter (for operating systems before Windows Vista)
- ROOT\SecurityCenter2 (Windows Vista and newer OS)

SELECT * FROM AntiVirusProduct

```
class AntiVirusProduct
{
    string  companyName;                // Vendor name
    string  displayName;                // Application name
    string  instanceGuid;               // Unique identifier
    boolean onAccessScanningEnabled; // Real-time protection
    boolean productUptoDate;            // Definition state
    string  versionNumber;              // Application version
}
```

AV identification

```
ab6ed3db3c243254294cfe431a8aeada28e5741dfa3b9c8aeb54291fddc4f8c3 (AhnLab)
b3fe0e3a3e3befa152c4237b0f3a96ffaa44a2d7e1aa6d379d3a1ab4659e1676 (AntiVir)
c0ffcaf63c2ca2974f44138b0956fed657073fde0adeb0b1c940b5c45e8a5cab (avast!)
249a90b07ed10bd0cd2bcc9819827267428261fb08e181f43e90807c63c65e80 (AVG)
4b650e5c4785025dee7bd65e3c5c527356717d7a1c0bfef5b4ada8ca1e9cbe17 (CA)
c8e8248940830e9f1dc600c189640e91c40f95caae4f3187fb04427980cdc479 (DoctorWeb)
97010f4c9ec0c01b8048dbad5f0c382a9269e22080ccd6f3f1d07e4909fac1a5 (F-PROT)
aa0ad154f949a518cc2be8a588d5e3523488c20c23b8eb8fafb7d8c34fa87145 (F-Secure)
333e0a1e27815d0ceee55c473fe3dc93d56c63e3bee2b3b4aee8eed6d70191a3 (G)
d4634c9d57c06983e1d2d6dc92e74e6103c132a97f8dc3e7158fa89420647ec3 (InternetSecurity)
977781971f7998ff4dbe47f3e1d679f1941b3237d0ba0fdca90178a15aec1f52 (Jiangmin)
f1761a5e3856dceb3e14d4555af92d3d1ac47604841f69fc72328b53ab45ca56 (Kaspersky)
a48be88bed64eff941be52590c07045b896bc3e87e7cf62985651bbc8484f945 (McAfee)
2bc42b202817bdab7d49506d291e3d9624ae0069087a8949c8fcb583c73772b1 (Norton)
0d21bd52022ca7f7e97109d28d327da1e68cc0bedd9713b2dc2b49d3aa104392 (Online)
f7d9ea7f3980635237d6ea58048057c33a218f2670e0ff45af5f4f670e9aa6f4 (Panda)
522e5549af01c747329d923110c058b7bb7e112816de64bd7919d7b9194fba5b (Rising)
4db3801a45802041baa44334303e0498c2640cd5dfd6892545487bf7c8c9219f (ThreatFire)
9e217716c4e03eee7a7e44590344d37252b0ae75966a7f8c34531cd7bed1aca7 (Trend)
e1625a7f2f6947ea8e9328e66562a8b255bc4d5721d427f943002bb2b9fc5645 (VirusBuster)
588730213eb6ace35caadcb651217bfbde3f615d94a9cca41a31ee9fa09b186c (ZoneAlarm)
b39be67ae54b99c5b05fa82a9313606c75bfc8b5c64f29c6037a32bf900926dd ()
a7f9b61169b52926bb364e557a52c07b34c9fbdcd692f249cd27de5f4169e700 ()
1ba035db418ad6acc8e0c173a49d124f3fcc89d0637496954a70e28ec6983ad7 ()
```

Emulator detection

Samples are looking for specific sandbox process names

```
if ( strstr(modulefilename, "klavme") )
{
  result = 1;
}
else if ( strstr(modulefilename, "myapp") )
{
  result = 1;
}
else if ( strstr(modulefilename, "TESTAPP") )
{
  result = 1;
}
else if ( strstr(modulefilename, "afyjevmv.exe") )
{
  result = 1;
```

Bitdefender

Kaspersky

Also Kaspersky:
lstcvix.exe
tudib.exe
izmdmv.exe
ubgncn.exe
jidgdsp.exe
evabgzib.exe
qzqjafyt.exe
cnyporqb.exe
…

Sample ID

All samples contain same looking ID:

- CSEC Slide: 08184
- Dino: 11173-01-PRS
- Bunny: 11206-01
- Babar: 11220-01 or 12075-01
- Casper 13001

Sample ID

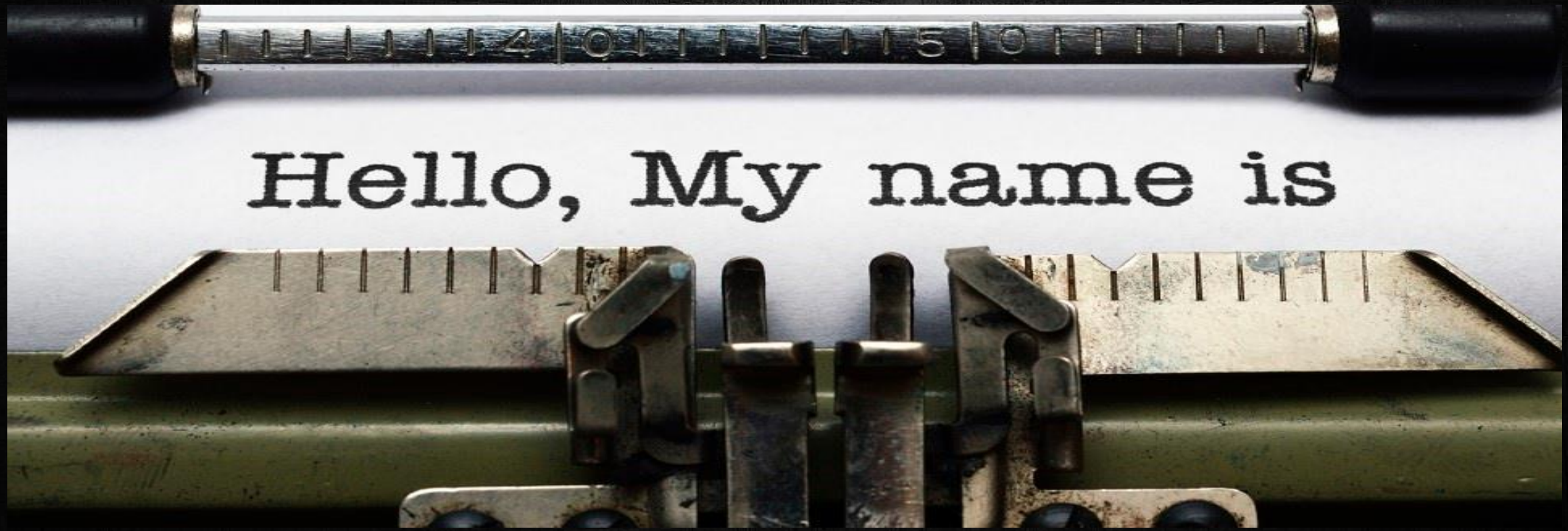
<speculation=on>

- CSEC Slide: 08184
- Dino: 11173-01-PRS
- Bunny: 11206-01
- Babar: 11220-01 or 12075-01
- Casper: 13001

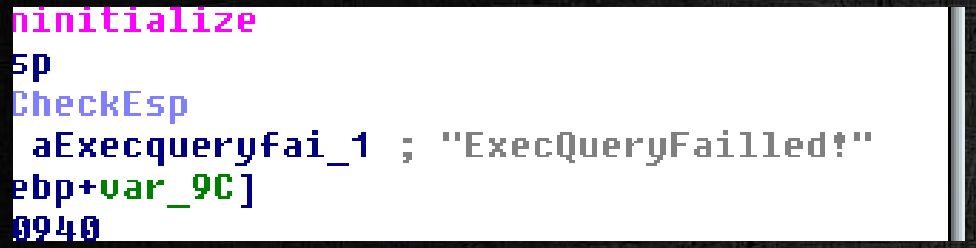
</speculation>

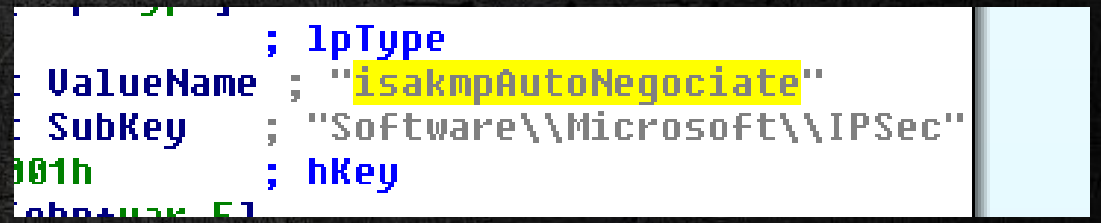
Internal naming convention

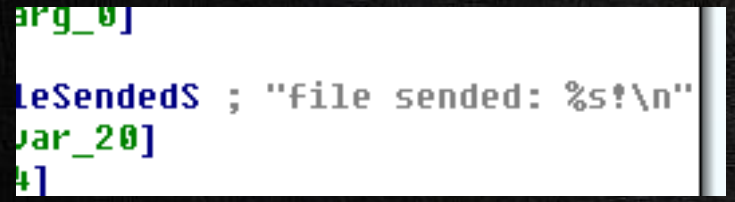
Of course the internal naming convention is a link too

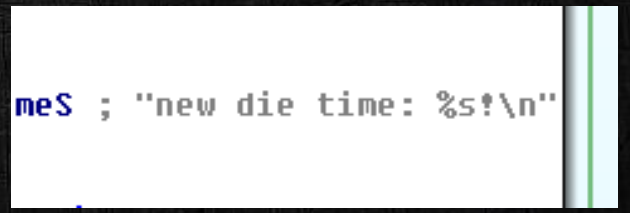


Really bad English usage









C&C sharing

Directory listing on horizons-tourisme.com:

```
./_vti_bin
./_vti_bin/index.html
./_vti_bin/_vti_msc
./_vti_bin/_vti_msc/d13
./_vti_bin/_vti_msc/d13/index_refresh.htm
./_vti_bin/_vti_msc/d13/index.html
./_vti_bin/_vti_msc/bb28
./_vti_bin/_vti_msc/bb28/_index.php
./_vti_bin/_vti_msc/bb28/storage
./_vti_bin/_vti_msc/bb28/storage/index.html
./_vti_bin/_vti_msc/bb28/index.html
./_vti_bin/_vti_msc/bb28/bbc.php
./_vti_bin/_vti_msc/bb28/config.inc
./_vti_bin/_vti_msc/tfc422
./_vti_bin/_vti_msc/tfc422/index.html
./_vti_bin/_vti_msc/index.html
```

Give me a “D” for Dino

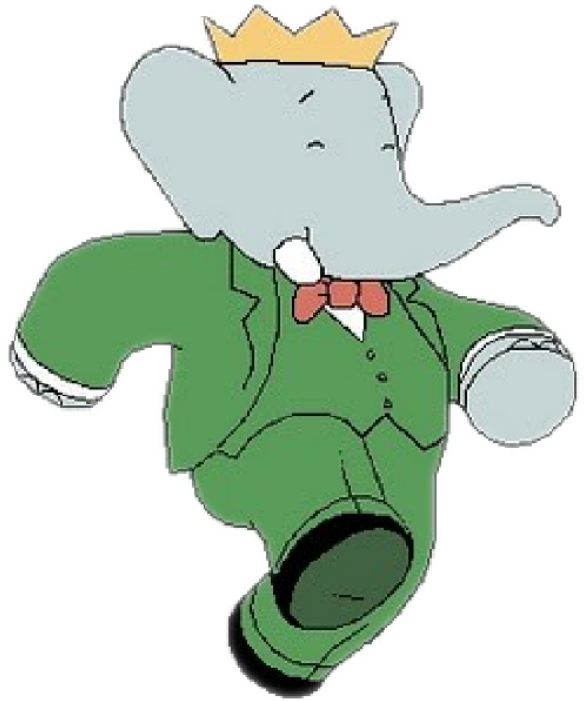
Give me 2 “B” for BaBar

Give me a “TFC” for TaFaCalou


China
Unit 61398
Cyber-
Crime
XSS
Attribution?

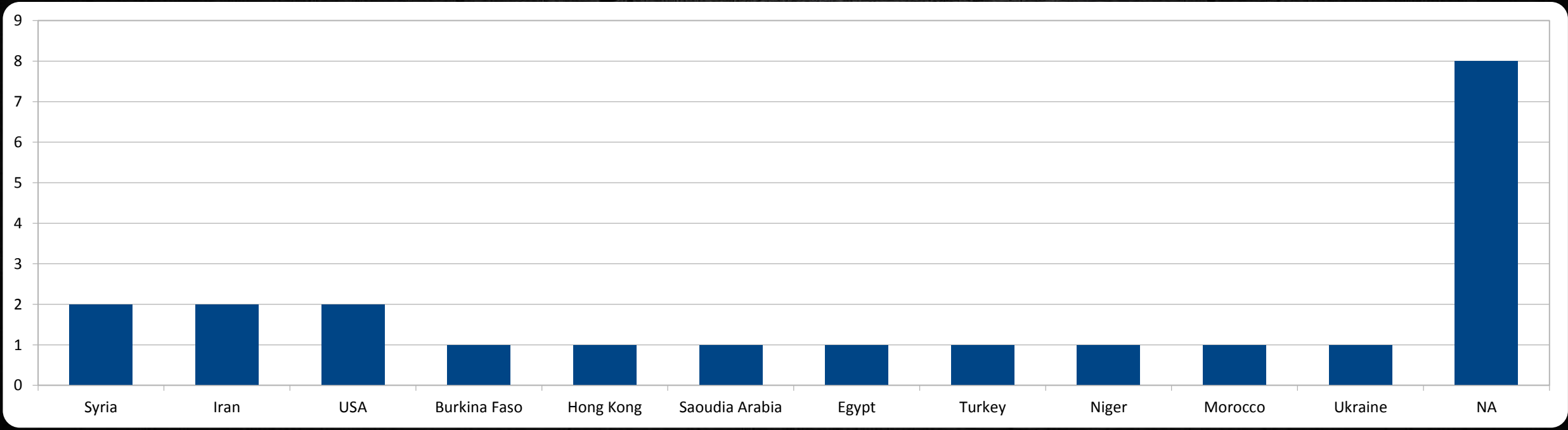
Attribution: Binary Artifacts

- ntrass.exe
  - DLL Loader uploaded to a victim as part of tasking seen in collection
  - Internal Name: Babar
  - Developer username: titi
- Babar is a popular French children's television show
- Titi is a French diminutive for Thiery, or a colloquial term for a small person



Analysis based on C&C

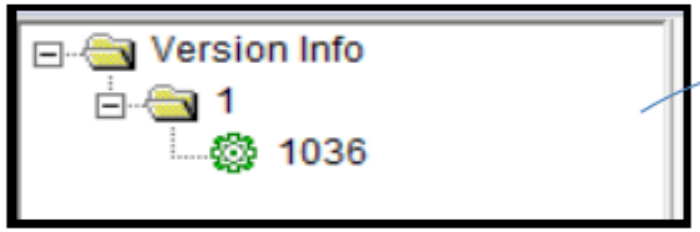
Compromised website (gov/university/company/...)
Often WordPress websites
Fake websites



Analysis based on C&C

A few French hints

```
%s  GET / HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, applic
Accept-Language: fr
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1 )
Accept-Encoding: gzip, deflate
```



French - Canada	fr	fr-ca	3084
French - Congo	fr		9228
French - Cote d'Ivoire	fr		12300
French - France	fr	fr-fr	1036
French - Luxembourg	fr	fr-lu	5132
French - Mali	fr		13324
French - Monaco	fr		6156

..\..\src\arithmetique\mpn\mul.c
..\..\src\arithmetique\printf\doprnt.c
..\..\src\arithmetique\mpn\tdiv_qr.c
..\..\src\arithmetique\mpn\mul_fft.c
..\..\src\arithmetique\mpn\get_str.c

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

SNOWGLOBE.

- CSEC assesses, with moderate certainty, SNOWGLOBE to be a state-sponsored CNO effort, put forth by a French intelligence agency

And FINALLY...

But attribution is hard.

Une des particularités de Casper est sa remarquable discrétion. « *Il adapte son comportement de façon très précise en fonction de l'antivirus qui s'exécute sur la machine où il est installé,* explique Joan Calvet, l'analyste québécois qui a réalisé l'étude d'Eset. *Par exemple, il préférera simplement ne pas contacter son contrôleur,*

renseignement français. "La France est aussi active que les gros ", avance la chercheuse australienne Marion Marschalek (Cyphort) au site Motherboard.

Joan Calvet
@joancalvet

Marion Marschalek
@pinkflawd

Paul Rascagnères
@r00tsbsd

Thank you!

Further Reading

- Babar Reversed https://drive.google.com/a/cyphort.com/file/d/0B9Mrr-en8FX4dzJqLWhDblhseTA/
- Bunny Reversed https://drive.google.com/file/d/0B9Mrr-en8FX4M2lXN1B4eElHcE0/view?usp=sharing
- Casper Reversed by Joan Calvet http://www.welivesecurity.com/2015/03/05/casper-malware-babar-bunny-another-espionage-cartoon/
- Blog on Babar http://www.cyphort.com/babar-suspected-nation-state-spyware-spotlight/
- Linking the Cartoon Malware to CSEC slides by Paul Rascagneres https://blog.gdatasoftware.com/blog/article/babar-espionage-software-finally-found-and-put-under-the-microscope.html
- Slides ‚TS/NOFORN' at Hack.lu2015 http://2014.hack.lu/archive/2014/TSNOFORN.pdf
- Slides on Snowglobe from CSEC http://www.spiegel.de/media/media-35683.pdf and http://www.spiegel.de/media/media-35688.pdf
- A cyberwarfare tale on nuclear matters by Matt Suiche http://www.msuiche.net/2015/03/09/did-alleged-dgse-used-stackoverflow-like-to-write-their-malwares/
- Animal Farm https://securelist.com/blog/research/69114/animals-in-the-apt-farm/
- !! COMING SOON !! Dino's analysis by Joan ☺

Hashes

Bunny:

- 3bbb59afdf9bda4ffdc644d9d51c53e7
- b8ac16701c3c15b103e61b5a317692bc
- c40e3ee23cf95d992b7cd0b7c01b8599
- eb2f16a59b07d3a196654c6041d0066e

Babar:

- 4525141d9e6e7b5a7f4e8c3db3f0c24c
- 9fff114f15b86896d8d4978c0ad2813d
- 8b3961f7f743daacfd67380a9085da4f
- 4582D9D2120FB9C80EF01E2135FA3515

NBOT:

- 8132ee00f64856cf10930fd72505cebe
- 2a64d331964dbdec8141f16585f392ba
- e8a333a726481a72b267ec6109939b0d
- 51cd931e9352b3b8f293bf3b9a9449d2

Casper:

- 4d7ca8d467770f657305c16474b845fe
- cc87d090a1607b4dde18730b79b78632

Dino:

- 30bd27b122c117fabf5fbfb0a6cdd7ee

Other:

- bbf4b1961ff0ce19db748616754da76e
- 330dc1a7f3930a2234e505ba11da0eea