[研究ノート]　新しいタイプの攻撃に対する出口対策としての認証プロキシの導入

岡本　剛

情報ネットワーク・コミュニケーション学科

An Egress Strategy using Authentication-Based Proxy for Advanced Persistent Threats

Takeshi OKAMOTO

Abstract

This report proposes an egress strategy using authentication-based proxy for advanced persistent threats. This is the 3rd strategy for blocking backdoor communication, which is one of egress strategies proposed by "A Guide of Design and Operation for a Countermeasure of Advanced Persistent Threats." This 3rd strategy uses an authentication-based proxy in order to discriminate malware communication and human communication. Even if a malware tries to communicate with a web server on Internet via a proxy server, the malware must be authenticated to pass through the proxy server. In that way, malware communication is blocked, while human communication is permitted. Blocking malware communication prevents the leak of confidential information and malware updates, etc.

Keywords: authentication-based proxy, advanced persistent threats, egress strategy, malware

1.　まえがき

情報処理推進機構（以下、IPA）では、特定の組織や個人を標的にした標的型攻撃と、言葉巧みにユーザを騙すソーシャルエンジニアリングを組み合わせた「新しいタイプの攻撃の対策に向けた設計・運用ガイド」[1]が公開されている。このガイドでは、マルウェアの多様化によるマルウェア検知の困難と、セキュリティの修正プログラムが公開されていない状態での脆弱性攻撃、すなわちゼロデイ攻撃が原因で、たとえ、内部ネットワークへの入口対策として、ファイアウォールや侵入防御システム、ウイルス対策ソフトといった対策を導入していても、「新しいタイプの攻撃」は、メールの添付ファイルとゼロデイ攻撃などを利用して、これらを突破して、内部ネットワークに侵入する。したがって、入り口対策ではなく、マルウェアが内部ネットワークから外部の指令サーバなどと通信するのを遮断する出口対策が求められている。その出口対策では、マルウェアに侵入（感染）しバックドアを仕掛けられたとしても、組織や個人が所有する機密情報を流出させたり、新しいマルウェアをダウンロードし感染させたりすることを防ぐ。つまり、出口対策で遮断されるべき通信は、バックドアと指令サーバの通信である。この通信を遮断することにより、たとえ、マルウェアに侵入されたとしても、機密情報の流出や新しいマルウェアのダウンロードなどを防ぐことができる。

IPAの設計・運用ガイドでは、図1のように、組織の内部ネットワークにプロキシを運用し、ネットワーク内部のユーザは、このプロキシを通してインターネット上のサーバと通信することを想定している。ファイアウォールでは、DMZとの通信（DNSやSMTPなど）と、プロキシとの通信のみを許可することとする。このネットワークでは、侵入後のマルウェアの通信は、次の4つのパターンに分類している。

1. プロキシを介さないHTTPに類する通信
2. プロキシを介さない独自のプロトコルの通信
3. プロキシを介して行うHTTPに類する通信
4. プロキシを介して行う独自のプロトコルの通信

これらのパターンの中で、1番目と2番目は、プロキシを介さないので、ファイアウォールで遮断される。つまり、プロキシを導入するだけで、これらの通信を遮断できる。一方、3番目と4番目は、ブラウザの通信を模倣しているため、プロキシを設置するだけでは、マルウェアの通信を遮断できない。そこで、設計・運用ガイドでは、マルウェアとユーザの通信を区別する方法として、次の二つの方式を提案している。

- JavaScriptやMETAタグを利用したリダイレクト方式

この方式は、HTTP 応答に対して、ブラウザは応答できるが、マルウェアは応答できないことを利用した方式である。具体的には、プロキシに JavaScript や META タグを利用したリダイレクト機能を実装し、リダイレクトに対する応答によりマルウェアと人間の通信を判別する。ただし、この方式は、プロキシにリダイレクト機能を実装する必要である。さらに、セッションを要求するような一部のサイトについては、通常の Web ブラウザからの閲覧に不具合が生じる可能性がある[2]。

- マルウェア固有の HTTP ヘッダなど、通信の特徴からバックドア通信を判別する方式

この方式は、マルウェアが行う HTTP を利用したバックドア通信とブラウザなどの人間による通信の違いを元に、マルウェアと人間の通信を識別する方式である。具体的には、HTTP ヘッダの正規 User-Agent をホワイトリスト化して識別する方法や、バックドアと指令サーバとのキープアライブ通信特性を登録するなどが挙げられている。ただし、いずれの方法もプロキシにこれらの機能を実装する必要がある。また、マルウェアがホワイトリスト化された User-Agent ヘッダを模倣したり、キープアライブを行わなかったりするなど、人間の通信を模倣されると識別が困難になる。

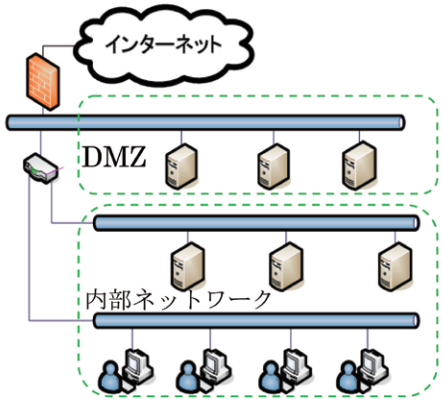

図 1 想定するネットワーク

本報告では、これら 2 つの手法とは別に、第 3 の対策を提案する。第 3 の方法では、認証プロキシを使うことにより、人間とマルウェアの識別を行う。人間しか知らない ID とパスワードを使うことにより、マルウェアと識別する。さらに、この方法は、プロキシの実装の 1 つ、Squid cache サーバ[3]であれば、標準で認証機能を備えているため、新たに実装する必要がない。

本報告では、第 2 章で認証プロキシの役割と設定内容について述べ、第 3 章では、提案手法の有効性を実験により評価する。第 4 章で、提案手法について考察を行い、第 5 章で結論を述べる。

2. 認証プロキシによる出口対策

認証プロキシとは、ユーザ認証を求めるプロキシである。プロキシ接続時に、ID とパスワードが求められる。マルウェアは、ID とパスワードを知らないため、認証プロキシを利用すれば、ユーザの通信を許可し、マルウェアの通信を遮断できる。

プロキシサーバには、いくつかの実装があるが、本研究では、認証機能を有する Squid cache サーバを利用する。Squid cache サーバのバージョンは、3.1.6 である。Squid cache サーバは、HTTP のベーシック認証やダイジェスト認証、マイクロソフト社の NTML 認証、LDAP 認証などの機能を有する。ベーシック認証は、平文のパスワードを BASE64 エンコードしただけであるため、プロキシの通信を盗聴すれば、簡単に ID とパスワードを見破れることから、ベーシック認証以外が望ましい。本報告では、チャレンジレスポンスの機能を備えた HTTP のダイジェスト認証を利用する。Squid cache サーバにおける HTTP ダイジェスト認証を行う設定（squid.conf の一部抜粋）を次に示す。

```
①auth_param digest program /usr/lib/squid3/digest_pw_auth -c
/etc/squid3/passwd
②auth_param digest realm Squid proxy-caching web server
③acl acl_proxy proxy_auth REQUIRED
④http_access allow localnet acl_proxy
```

①は、/etc/squid3/passwd にある ID とパスワードに基づいて、ダイジェスト認証を行う設定である。/etc/squid3/passwd は htdigest コマンドにより生成した。②は、ID とパスワードの入力するためのダイアログに表示されるメッセージを「Squid proxy-caching web server」に設定している。③と④により、/etc/squid3/passwd に登録されたすべてのユーザに対して認証が要求されるように設定する。

Squid cache サーバによる認証は、アプリケーション毎に求められる。ブラウザであれば、ブラウザを起動して、ブラウザでホームページを閲覧しようとしたときに ID とパスワードが求められる。なお、ブラウザの ID とパスワードの記憶機能を使えば、ブラウザ起動時だけ ID とパスワードが求められ、それ以降は、ブラウザに記憶された ID とパスワードを利用して、自動的に認証が行われる。一方、マルウェアが HTTP の通信を模倣して、プロキシを通過しようとしたとき、マルウェアに対して、認証が求められる。このとき、マルウェアは、ID とパスワードを知らないため、マルウェアの通信は、このプロキシで遮断される。マルウェアが、ID とパスワードを窃取するために、ネットワークを盗聴する機能を持っていたとしても、HTTP のダイジェスト認証は、チャレンジレスポンス方式であるため、認証のたびに、レスポンスが変更されるので、マルウェアが認証を成功させることは困難である。

3. 認証プロキシによる出口対策の実験

本実験では、内部ネットワーク内のコンピュータが、脆

弱性攻撃により管理者権限が奪取され、インターネット上のコンピュータからマルウェアをダウンロードして、それを実行し、マルウェアに感染する状況を想定する。次に、提案手法により、マルウェアのダウンロードを未然に防げることを確認する。具体的には、Adobe Acrobat で見つかった脆弱性CVE-2009-0927への攻撃が仕込まれたPDFファイルを標的（特定のユーザ）にメールで送り、標的が騙されて、PDFファイルを開き、管理者権限を奪取され、インターネット上のコンピュータからマルウェア（実験では電卓）をダウンロードされ、マルウェア（電卓）を実行された場合を想定する。

実験では、PDFファイルは、Metasploit（バージョンは4.1.2）[4]により生成した。マルウェアのダウンロードと実行は、Metasploit によって仕込まれたシェルコード（download_exec）によって行われる。そのシェルコードでは、Windows の API 関数 URLDownloadToFile により http://www.nw.kanagawa-it.ac.jp/~take4/calc.exe からファイル（calc.exe）をダウンロードし、API 関数 WinExec によりダウンロードしたファイルを実行する。URLDownloadToFile 関数は、インターネットオプションで設定されたプロキシを利用してファイルをダウンロードする。その設定が有効の場合、プロキシを利用してファイルをダウンロードし、その設定が無効の場合は、プロキシを介さないでファイルをダウンロードする。これにより、プロキシを介して行う通信を模擬できる。

実験では、プロキシに認証機能を設定しない場合と、認証機能を設定した場合（プロキシを介して行う通信）について、ネットワークトラフィックを調べた。

3.1 プロキシに認証機能を設定しない場合

プロキシに認証機能を設定しない場合は、プロキシで認証する必要がないので、マルウェアの通信は遮断されることなくダウンロードが完了し実行され、マルウェアが起動する。そのネットワークトラフィックの先頭部分を抜粋したデータを図2に示す。

4番目のパケットでダウンロードのリクエスト（GET /~take4/calc.exe HTTP/1.1）が送られ、6番目以降のレスポンスとして、マルウェア（電卓）のファイルが送られている。この通信が完了したら、マルウェア（電卓）が起動した。

3.2 プロキシに認証機能を設定した場合

プロキシに認証機能を設定した場合は、プロキシで認証する必要があるので、プロキシで ID とパスワードが要求される。マルウェアには、認証する機能がなく、マルウェアは ID とパスワードを知らないため、マルウェアの通信は遮断される。そのネットワークトラフィックを図3に示す。

4番目のパケットでダウンロードのリクエストが送られ、8番目のパケットで認証が要求（HTTP/1.0 407 Proxy Authentication Required (text/html)）され、その後、通信が途絶えていることがわかる。つまり、認証プロキシによって、マルウェアの通信が遮断されたことが確認できる。Squid cache サーバには、ログに「1321073790.930 0 192.168.35.105 TCP_DENIED/407 4238 GET http://www.nw.kanagawa-it.ac.jp/~take4/calc.exe - NONE/- text/html」が出力され、脆弱性攻撃を受けた可能性のあるコンピュータの IP アドレスがわかる（下線部の IP アドレス）。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.35.105	202.250.71.135	TCP	62	zented > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
2	0.000743	202.250.71.135	192.168.35.105	TCP	62	http > zented [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M
3	0.000771	192.168.35.105	202.250.71.135	TCP	54	zented > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.016195	192.168.35.105	202.250.71.135	HTTP	265	GET /~take4/calc.exe HTTP/1.1
5	0.016874	202.250.71.135	192.168.35.105	TCP	60	http > zented [ACK] Seq=1 Ack=212 Win=6432 Len=0
6	0.017332	202.250.71.135	192.168.35.105	TCP	308	[TCP segment of a reassembled PDU]
7	0.017574	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
8	0.017593	192.168.35.105	202.250.71.135	TCP	54	zented > http [ACK] Seq=212 Ack=1715 Win=64240 Len=0
9	0.018867	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
10	0.018875	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
11	0.018892	192.168.35.105	202.250.71.135	TCP	54	zented > http [ACK] Seq=212 Ack=4635 Win=64240 Len=0
12	0.018987	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
13	0.019971	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
14	0.019989	192.168.35.105	202.250.71.135	TCP	54	zented > http [ACK] Seq=212 Ack=7555 Win=64240 Len=0
15	0.020213	202.250.71.135	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]

図2　プロキシに認証機能を設定しない場合のネットワークトラフィック

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.35.105	192.168.35.5	TCP	62	caiccipc > ndl-aas [SYN] Seq=0 Win=64240 Len=0 MSS=1460
2	0.000213	192.168.35.5	192.168.35.105	TCP	62	ndl-aas > caiccipc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=
3	0.000244	192.168.35.105	192.168.35.5	TCP	54	caiccipc > ndl-aas [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.008052	192.168.35.105	192.168.35.5	HTTP	392	GET http://www.nw.kanagawa-it.ac.jp/~take4/calc.exe HTT
5	0.008266	192.168.35.5	192.168.35.105	TCP	60	ndl-aas > caiccipc [ACK] Seq=1 Ack=339 Win=6432 Len=0
6	0.008663	192.168.35.5	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
7	0.008705	192.168.35.5	192.168.35.105	TCP	1514	[TCP segment of a reassembled PDU]
8	0.008712	192.168.35.5	192.168.35.105	HTTP	1372	HTTP/1.0 407 Proxy Authentication Required (text/html)
9	0.008745	192.168.35.105	192.168.35.5	TCP	54	caiccipc > ndl-aas [ACK] Seq=339 Ack=4239 Win=64240 Len
10	0.012328	192.168.35.105	192.168.35.5	TCP	54	caiccipc > ndl-aas [RST, ACK] Seq=339 Ack=4239 Win=0 Le

図3　プロキシに認証機能を設定した場合のネットワークトラフィック

一方、ユーザによる通信は、プロキシから認証が要求されたときに、ユーザが正しいIDとパスワードを入力すれば、通信は許可され、プロキシ経由で通信が再開される。

次に、プロキシに認証機能を設定した状態で、マルウェアがブラウザの脆弱性を攻撃し、ブラウザを乗っ取った場合について考える。実験では、Microsoft Internet Explorerで見つかった脆弱性CVE-2012-1889を攻撃するウェブページを用意し、ユーザが脆弱性のあるInternet Explorerでそのウェブページを閲覧した場合を想定する。攻撃に用いるシェルコードは前ページの実験でPDFファイルに埋め込んだものと同じである。実験の結果、シェルコードがすべて正しく実行され、最後に、マルウェア（電卓）が起動した。この実験結果から、マルウェアはすでに認証されたブラウザの認証情報を利用してプロキシの認証を通過することを確認した。

4. 考察

4.1 IDとパスワード

認証プロキシでは、ユーザにIDとパスワードの記憶が求められる。プロキシに設定されるIDとパスワードは、マルウェアと人間を識別するために使うため、必ずしも、ユーザ単位である必要はなく、組織やグループ単位でもよい。IDとパスワードを要求するだけで、シェルコードに、IDとパスワードを窃取する機能と認証情報を送信する機能を追加しなければならないため、シェルコードの作成が困難になると予想される。

4.2 その他の認証

ユーザが多数の場合は、認証情報をLDAPサーバに保存し、LDAP認証を使う方法がある。組織内でLDAP認証をすでに使っている場合には、それを利用することができる。同様に、NTLM認証も同じことができる。LDAP認証とNTLM認証では、認証情報がSquid cacheサーバとは別のサーバで管理できることから安全性をさらに向上させることができる。

4.2 認証プロキシの弱点

ブラウザに脆弱性があり、マルウェアがその脆弱性を不正利用して、そのブラウザを乗っ取った場合には、マルウェアはそのブラウザになりすますことができる。このとき、マルウェアはブラウザが認証に成功した認証情報を利用して通信するため、プロキシで遮断されない。

しかし、新しいタイプの攻撃でよく利用されるドロッパーやバックドアには、認証プロキシは有効に作用する。ブラウザが乗っ取られた後に、ドロッパーやバックドアがインターネット上のホストからダウンロードされインストールされたとしても、これらのマルウェアは、認証情報を有しないため、プロキシで遮断される。つまり、たとえ、ドロッパーやバックドアがインストールされても、これらのマルウェアにより、新しいマルウェアがインストールされたり、指令サーバと通信し、指令サーバの指示によって機密情報の流出などの不正行為を行ったりすることはできない。

5. まとめ

本報告では、IPAが提供する「新しいタイプの攻撃の対策に向けた設計・運用ガイド」で提案されている出口対策の1つ、マルウェアのバックドア通信の遮断について、第3の方法を提案した。第3の方法は、マルウェアと人間の通信を識別することを目的として、プロキシに認証機能を導入することである。マルウェアは、たとえ、プロキシを介した通信を行ったとしても、プロキシを通過するには、認証に成功しなければならない。すなわち、IDとパスワードの入力が必要になる。これにより、マルウェアの通信を遮断し、人間による通信を許可することが可能になる。実験では、Metasploitを利用して、脆弱性攻撃とマルウェアのダウンロードを行うPDFファイルを作成して、管理者権限を奪取されたコンピュータが認証プロキシでダウンロードが遮断されることを確認した。しかし、脆弱性攻撃によりブラウザが乗っ取られた場合には、プロキシで遮断できないことを確認した。ブラウザが乗っ取られた場合には、乗っ取られたブラウザの通信は遮断できないが、そのブラウザでダウンロードしてインストールされたマルウェア（ドロッパーやバックドアなど）は、認証情報を有しないため、プロキシによって遮断される。つまり、この認証プロキシにより、機密情報の流出や、マルウェアのアップデートなどを防ぐことができる。

参考文献

[1] 情報処理推進機構セキュリティセンター：「新しいタイプの攻撃」の対策に向けた設計・運用ガイド，情報処理推進機構，
http://www.ipa.go.jp/security/vuln/newattack.html
[2] 相馬 基邦：HTTPベースでバックドア通信を遮断する，ITpro セキュリティ，
http://itpro.nikkeibp.co.jp/article/COLUMN/20110930/369699/?ST=security&P=2
[3] Squid cache: squid : auth_param configuration directive,
http://www.squid-cache.org/Doc/config/auth_param/
[4] Metasploit, http://metasploit.com/