2014年度 SWAN Working Group 活動報告

宮本大輔 (daisu-mi@nc.u-tokyo.ac.jp)　藤原 寛高 (fujiwara.hirotaka.ex3@is.naist.jp)
Gregory Blanc (gregory.blanc@telecom-sudparis.eu)　門林雄基 (youki-k@is.naist.jp)

2014年12月15日

目 次

1 はじめに

SWAN (Security for Web 2.0 Application) WG では，悪意あるウェブサイトの動向を観測し検討している．ウェブを介した攻撃にはその攻撃空間が広いという特徴があり，本研究グループはその広い特性に対応した研究を行なっている．これまでの活動としては，エンドユーザの認知能力に合わせたフィッシングサイト解析や脆弱性を持つウェブ 2.0 のアプリケーションを WIDE メンバーに提供する試み，悪意あるウェブサイトによく見られる難読化された JavaScript の構造に着目した解析，PC だけではなく Android などで動くマルウェアの解析技術のハンズオンなどが挙げられる．

今年度は，多くの WIDE プロジェクトメンバーが参加している日欧協調によるマルチレイヤ脅威分析およびサイバー防御の研究開発（NECOMA [1]）と SWAN WG は連携した研究を行い,「アドレスバーを目視で確認する」ことをエンドユーザに習慣として身につけさせる取り組み，2014 年 WIDE 秋合宿で行った被験者実験，そして難読化の特徴を利用したドライブバイダウンロード攻撃検知手法の研究開発を実施した．

以降 SWAN WG の本年度の主な活動について報告する．

2 アドレスバーを目視で確認する習慣を身につけさせる取り組み

フィッシングサイト対策は，大きく 3 通りの手法に分類される．第一に，エンドユーザに対してウェブサイトの真贋判定に必要な知識を教育する手法がある．URL や SSL の鍵アイコンなどの情報に対する知識不足はフィッシング問題の原因として挙げられることも多く，教育に必要な様々な教材，授業様式が研究開発されている．また，エンドユーザが重要な情報を見落としにくくするためのユーザインタフェースの開発も行われている．SSL の鍵アイコンより，EV-SSL のアドレスバーが緑色に変わる仕組みの方がユーザの注意喚起を促せるであろうため，この手法も有力である．フィッシングサイトか否かを判定し，エンドユーザに通知する仕組みも行われており，我々の研究グループも機械学習を用いたフィッシングサイトの検知手法 [1] や，各エンドユーザの判断と機械学習結果を融合する検知手法 [2] に取り組んでいる．

[1] http://www.necoma-project.jp/

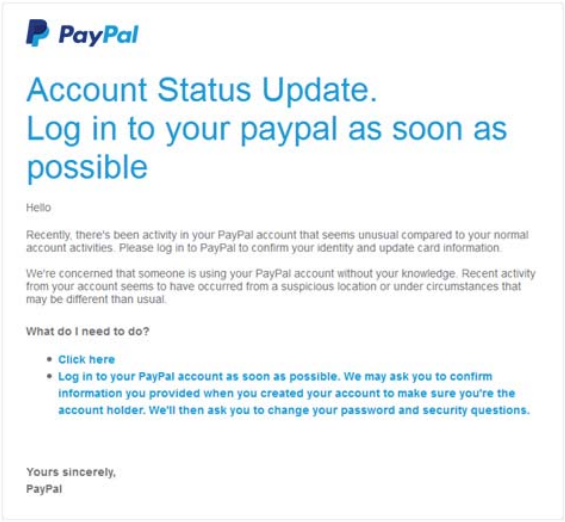

図 1: フィッシングメールの例

さて，フィッシングサイトを閲覧するエンドユーザは，おそらくフィッシングメール等によって誘導されてきたものと思われる．このようなメールには，ユーザを心理的に揺さぶるような文言が書かれている特徴がある．例えば図 1 に示すフィッシングメールは，金融機関を装って「あなたのアカウントが凍結されました」などの文言が書かれている．ここで問題となるのは，エンドユーザにとって注意深くウェブサイトを閲覧することよりも，事態を確認するなどの心理が優先され，ユーザがメールに示されるウェブサイトに誘導され，個人情報を入力するような場合である．URL や SSL の鍵アイコンといった知識やインタフェースは重要であるが，エンドユーザがその情報を閲覧せずに，コンテンツだけで真贋判定してしまった場合，教育やインタフェース開発によって得られた対策効果が効力を発揮しない．

この問題への対策として，我々は URL や SSL の鍵アイコンを確認するような習慣をエンドユーザに身につけさせる方法について考える．習慣は，ある種の条件反射的な活動であり，無意識の動作であるとされる．たとえエンドユーザの心理状態が注意深くウェブサイトを閲覧するような状態でなかったとしても，習慣によって URL や SSL の鍵アイコンを確認することにより，フィッシングサイトに気づく可能性が高まるのではないかと考えた．

そこで，エンドユーザが URL や SSL の鍵アイコンが表示されるアドレスバーを閲覧することの有効性の評価を行う．そして，エンドユーザがアドレスバーを閲覧しない限り情報が入力できなくなるブラウザ拡張 *EyeBit* を作成し，この効果を測定する．

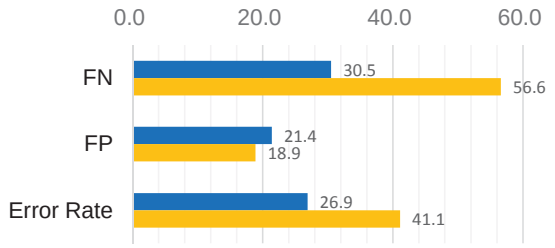

図 2: アドレスバーを見た場合，見ない場合の判定精度

2.1 アドレスバーを閲覧する効果

エンドユーザの視線情報を収集するため，2013 年 11 月から 2014 年 2 月までの期間，東京大学の構内掲示板にて被験者を募集した．応募があった被験者には，実験の目的として，「セキュリティ技術の研究開発を目的としたウェブサイトを閲覧した際のエンドユーザの挙動の観測」であることを説明し，作業内容として「ウェブサイトの画面を閲覧してもらい，正規のサイトか，あるいは偽サイトかを判定していただきます．またその際に判定基準をアンケート形式でお答えいただきます」と説明した．この実験に関する個人に属する情報として，性別，年代（10 代，20 代，30 代……60 代以上），ウェブサイトを見た際の判定結果（正規サイト，偽サイト），判定基準（ウェブサイトのコンテンツ，アドレスバーに表示される URL，ブラウザの表示するセキュリティ情報，その他，の 4 択），視線の動き（判定時の目線の動き）を取得し，個人を特定可能な情報は記録しないことを説明した．また，取得した情報を日欧の研究コンソーシアム間で共有すること，またセキュリティ技術の研究開発を目的とした場合に配布することについて許諾を得た．さらに，欧州における忘れられる権利について対応すべく，個人を特定可能な情報は記録しないものの，将来的に個人情報の解釈が変わる可能性も考慮し，我々の取得した個人に属する情報について被験者らが消去依頼を行える旨を説明した．ただし，我々の保存するデータからは個人を特定可能な情報は含まれないため，十分な長さの乱

表 1: EyeBit の評価に用いたウェブサイト

#	ウェブサイト	フィッシングか否か	言語	備考
1	Yahoo	yes	JP	`dmiurdrgs.cher-ish.net`, once reported as a phishing site
2	PayPal	no	EN	EV-SSL
3	eBay	yes	EN	`signin-ebay.com`, similar to legitimate URL `signin.ebay.com`
4	DMM	no	JP	SSL
5	Amazon	yes	EN	`www.importen.se`, once reported as a phishing site
6	Bank of America	no	EN	EV-SSL
7	Facebook	no	JP	SSL
8	Square Enix	yes	JP	`hiroba.dqx.jp...`, similar to legitimate URL `hiroba.dqx.jp`
9	Twitter	yes	JP	`twittelr.com`
10	Google	no	JP	SSL
11	Battle.net	no	EN	EV-SSL
12	Sumitomo Mitsui Card	yes	JP	`www.smcb.card.com....`, similar to legitimate URL `www.smbc-card.com`

数の文字列を発行して相互に保管しておき，その文字列をもって個別の情報を消去できるようにした．実験参加者は 25 人であり，そのうち 2 名は実験シナリオに不備があり，視線情報を正しく判別できていなかった．残りの 23 名についてそのうち 20 人が男性，3 人が女性であった．また，22 人が 20 代であり，残りの 1 人が 30 代であった．実験の際に用いたフィッシングサイトについては文献 [3] を参照されたい．

図 2 にアドレスバーを見た場合，見ない場合の判定制度を示す．延べ 331 回のアドレスバーを目視した回数のうち，誤判定があったのは 89 回であった．フィッシングサイトに限定していえば 200 回のアドレスバーを目視した回数のうち 61 回が誤判定であり，残りの 131 回の正規サイトにおいてアドレスバーを目視した場合の誤判定は 28 回であった．従って，エラー率，False Positive 率，False Negative 率はそれぞれ 26.9%，21.4%，30.5%となる．反対にアドレスバーを見ない場合は，41.1%（129 回中 53 回），18.9%（53 回中 10 回），56.6%（76 回中 43 回）であった．False Positive 率ではごくわずかにアドレスバーを見ない場合が低くなっているが，フィッシングサイトのコンテンツは正規サイトと見た目が区別しにくいため，アドレスバーを見ない限り False Positive 率が高くなっている．結論として，アドレスバーを見ることは効果的であると考えられる．

2.2 EyeBit による習慣付けの効果

アドレスバーを見る習慣をエンドユーザに取得させるにはどうすればよいか．単純な方法ではあるが，その行動を強制し，反復することによって効果が得られるのではないかと考える．そこで，視線追跡カメラを用いて視線位置を特定し，アドレスバーを見ないかぎりウェブサイトの入力フォームを利用不可能にする仕組みを考えることとした．

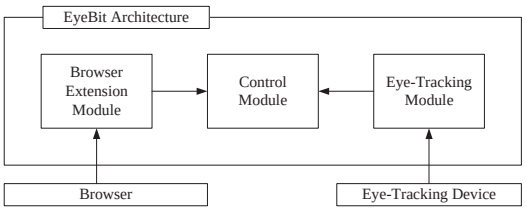

図 3: EyeBit の構成図

図 3 は，我々が提案する仕組みである EyeBit のアーキテクチャを示す．EyeBit はブラウザ拡張モジュール，視線追跡モジュール，制御モジュールの 3 つのコンポーネントから構成される．ブラウザ拡張モジュールはウェブサイト閲覧時に全ての入力フォームを利用不可能にし，制御モジュールからの指示があるまで無効化し続ける．視線追跡モジュールは，視線追跡デバイスと連携し，エンドユーザの目線位置の計測を行う．アドレスバーを閲覧していることが確認された場合，視線追跡デバイスは制御モジュールにメッセージを伝達し，制御モジュールがブラウザ拡張モジュールに対し入力フォームを利用可能にするよう指示を出す．

EyeBit の視線追跡モジュールは Go 言語で，ブラウザ拡張モジュールは JavaScript で Google Chrome ブラウザ拡張として実装し，オープンソースコードとして公開している [2]．視線追跡カメラには EyeTribe Tracker [3] を用いた．なお，EyeTribe には SDK が用意されており，JSON フォーマットでの視線位置情報を獲得することができる．

本システムを用い，奈良先端科学技術大学院大学及

[2] `https://github.com/necoma`
[3] `https://theeyetribe.com`

表 2: 判定結果

#	A_1	A_2	A_3	A_4	A_5	B_1	B_2	B_3	B_4	B_5
1	F				F		F	F		
2										
3	F				F		F	F	F	
4	F		F					F		
5									F	
6										
7			F							
8										
9					F	F				
10		F	F							
11	F	F	F					F		
12										

び東京大学から 10 人の被験者を募集し実験を行った．実験は以下に示す 5 つの手順によって構成される．実験の際に用いたウェブサイトは表 1 の通りである．なお，フィッシングサイトはテスト空間上に再現し，被験者以外のユーザが閲覧できないよう設定を行った．

フェーズ 1: ウェブサイト 1-4 の判別 10 人の被験者はそれぞれ表 1 のウェブサイト 1-4 を閲覧する．正規サイトだと思った場合は，被験者はユーザ名に仮の人格である ”john” を入力し，フィッシングサイトだと思った場合はそのサイトを離れることによって表明してもらう．

フェーズ 2: 教育時間 被験者に，フィッシングサイトの判別方法について教えるべく，URL とは何か，SSL とは何か，EV-SSL とは何かを説明する．

フェーズ 3: ウェブサイト 5-8 の判別 10 人の被験者はそれぞれ表 1 のウェブサイト 1-4 を閲覧する．ここで，10 人のうち 5 人は EyeBit を用いながら判別を行う．残りの 5 人は何も用いずに判別を行う．

フェーズ 4: 休憩時間 10 人の被験者は 1 時間の休憩を行う．

フェーズ 5: 休憩時間 10 人の被験者は何もつけずに表 1 のウェブサイト 9-12 を閲覧する．この手順の意図は，EyeBit を用いた場合と用いなかった場合で，判定にどのような差が現れるかを調べるものであった．

解答結果を図 2 に示す．A_{1-5} がフェーズ 3 において EyeBit を用いた被験者，B_{1-5} が用いなかった被験者である．記号 ”F” が表示されているところは，被験者がそのサイトの判別を誤ったことを示し，空白は正しく判別したことを示す．被験者 A_1，A_5，B_2 及び B_3 は，フェーズ 1 において全てのサイトを正規サイトと答えており，フィッシングサイトであるウェブサイト 1 と 3 を正しく判定できてないことから，おそらく初心者であろうと思われる．教材を与えた直後のフェーズ 3 では被験者グループ A も B も正しい判定ができる傾向が見られ，1 時間休憩した後のフェーズ 5 ではやや教育効果が減衰したようにも思われる．とりわけ被験者 A_2，A_3 が正規のウェブサイトをフィッシングサイトであると懐疑的に判断する傾向が見られた．フェーズ 5 を比較すると，EyeBit による教育効果は判定に補正を与えてないのではないかと思われる．また，いずれのユーザもアドレスバーを見て判断することが確認された．

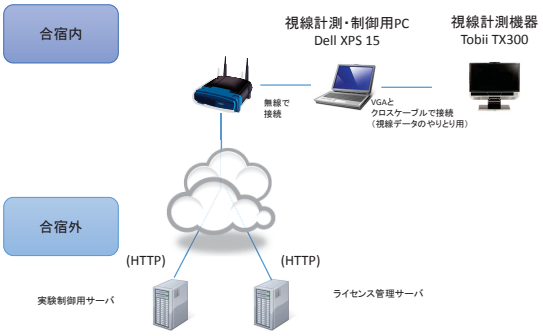

図 4: 合宿における実験構成図

この結果を受けた 1ヶ月後に，初心者と思われる 4 人の被験者に再度追試を行った．追試では表 1 のウェブサイト 1-12 について，EyeBit を用いずにフィッシングサイトか否かを判定してもらうという単純な内容であった．実験では A_1，A_5，B_2 の被験者についてはアドレスバーを閲覧する傾向が見られたが，B_3 の被験者はアドレスバーを見る習慣が損なわれていた．

この結果から単純に EyeBit がエンドユーザにアドレスバーを見る習慣付けがなされたとは断定はできないが，被験者数の増加などのさらなる追試を行い，有効性を調査する予定である．

3 2014年WIDE秋合宿参加者によるフィッシングサイト判別実験

2.1 節に述べた被験者を募った実験を，2014 年 WIDE 秋合宿においても同様に行った．実験の構成図を図 4

に示す．被験者は制御用 PC が表示するウェブサイトの画像を視線計測機器を通して閲覧する．今回の実験で用いた Tobii TX 300 [4] はヘッドマウント型の機器ではなくモニタ型の機器であり，被験者は通常の PC 画面を見るようになっている．視線計測・制御用 PC は，Tobii TX300 用の視線追跡を行い，なおかつフィッシングサイトの画像を表示し，アンケートを集計するウェブアプリケーションにエンドユーザを誘導する役割を担う．なお，視線計測・制御用 PC は，IPv6 のみを利用する合宿ネットワークを利用していたが，視線追跡を行うソフトウェアは外部のライセンス管理サーバとライセンス認証の通信を行う必要があった．ライセンス認証サーバは IPv4 アドレスのみが利用可能であるように思われたが，NAT64/DNS64 環境により通信が行えることを確認している．また，実験制御用サーバはエンドユーザにアンケート画面を表示する役割を担っている．

実験会場に来られた各被験者の方は 34 名，そのうち男性が 31 名，女性が 3 名であった．年代は，10 代が 1 名，20 代が 21 名，30 代が 8 名，40 代が 4 名である．2.1 節に述べた被験者実験と同様の実験を行った．

本実験による平均エラー率は 21.6%，False Positive 率は 27.9%，False Negative 率は 17.4%であった．特に WIDE 合宿では実験参加者がネットワークを研究される方が多いことは想像されうるため，2.1 節に述べた被験者実験とくらべて偏りがあることは予想されるところである．有意水準を 5%とした T 検定を行った所，エラー率は有意差がある（$p = 0.017 < 0.05$）ことが観測された．ただし，False Negative 率が比較的低い反面，False Positive 率が高い傾向にある．データセットにおけるフィッシングサイトが 12 サイト，正規サイトが 8 サイトであることから，比較的フィッシングサイトだという解答を続けることによりエラー率は改善され得る．フィッシングサイトよりの解答を続ける理由の源泉としては,「判定困難な場合はフィッシングサイトである」という心理や，正規サイトであっても信用できないものは信用できないという心理が働いた可能性が考えられる．

意思決定の際に用いた根拠については，URL やブラウザの表示するセキュリティ情報に基づいて判定した場合のエラー率が最も低い（8.8%）ことに対し，コンテンツのみに頼った判断をした場合は高くなっている（55.6%）．意思決定時における内在する精神状態や視線情報の分析は今後の課題である．

[4] http://www.tobii.com/ja-JP/eye-tracking-research/japan/products/hardware/tobii-tx300-eye-tracker/

表 3: アンケートによる選択結果

コンテンツ	URL	セキュリティ情報	エラー率	FP 率	FN 率
v			55.6%	45.5%	64.1%
	v		14.1%	38.7%	06.5%
		v	32.0%	53.3%	00.0%
v	v		25.3%	12.5%	40.9%
v		v	12.5%	14.3%	00.0%
	v	v	0.88%	14.6%	02.6%
v	v	v	24.0%	26.3%	16.7%

4 難読化の特徴を利用したドライブバイダウンロード攻撃検知手法の提案

ドライブバイダウンロード攻撃はユーザを悪意のあるサイトへ誘導し，ウェブブラウザやそのプラグインなどの脆弱性を悪用してユーザの端末にマルウェアを感染させる攻撃である．我々は，良性の難読化と悪意のある難読化について，文字列の長さや使用される文字といった特徴の調査を行った．また, この攻撃で利用される難読化の特徴から，ドメイン情報を利用した攻撃検知について提案する．

4.1 難読化について

難読化は様々な分野で利用されており，一般的には他者から自らのコードを守るために難読化処理を行う．難読化には複数の手法が存在しており，ランダム難読化，データ難読化，エンコード難読化に分類することが可能である．図 5 の（a）がオリジナルのコードとなっており, (b) , (c) , (d) は各難読化を施したものである．ランダム難読化では図 5 の（b）のように変数名や関数名といったものを変更することで，変数や関数の動作をわかりにくくすることが可能となる．データ難読化では図 5 の（c）のように意味のある文字列を複数に分割し，eval や document.write といった関数を利用することで，意味のある文字列に戻す．データ難読化には文字列を分割する方法の他に，順に変数に代入していくことで意味のある文字列にする方法も存在する．エンコード難読化では図 5 の（d）のように ASCII コードや Unicode で書かれた文字列を unescape などを利

```
function myAlert(txt){
    alert(txt);
}
var string=“Hello World!!”;
myAlert(string);
```

(a)

```
function   cd(ab){
    alert(ab);
}
var ok=“Hello World!!”;
  cd(ok);
```

(b)

```
var co = “ert(txt)“;
var pg = “ello World!“;
var am = “functi“;
var qf = “tring=\”H“;
var jl = “ing)“;
var ne = “xt);}var s“;
var rh = “l\”;myA“;
var wb = “on myAl“;
var ik = “lert(str“;
var sd = “{alert(t“;
eval(am+wb+co+sd+ne+qf+pg+rh+ik+jl);
```

(c)

```
unescape(%66%75%6e%63%74%69%6f%6e%20%6d%79%41%6c
%65%72%74%28%74%78%74%29%7b%61%6c
%65%72%74%28%74%78%74%29%3b%7d
%76%61%72%20%73%74%72%69%6e%67%3d%e2%80%9c
%48%65%6c%6c%6f%20%57%6f%72%6c
%64%21%21%e2%80%9d%3b%6d%79%41%6c
%65%72%74%28%73%74%72%69%6e%67%29%3b);
```

(d)

図 5: （a）オリジナル （b）ランダム難読化 （c）データ難読化 （d）エンコード難読化

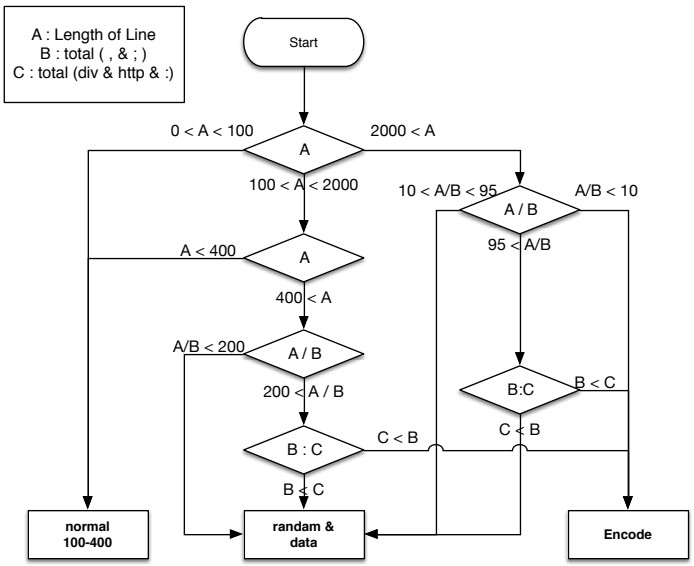

図 6: 文字列の長さを利用した難読化の分類

用して解除する．他にデコード用の関数を用意し，難読化の解除を行うものも存在する．難読化ではこれらの手法を組み合わせたものが一般的に利用されている．

今回は良性と悪性の難読化について，Alexa と D3M データセットおよび Malwr.com を用いて調査を行った．難読化の調査にはトップページと読み込まれる JavaScript ファイルを用いて，文字列の長さや使用されている文字，その頻度などについて静的解析を行い，考察した．Alexa に掲載されているサイトを正規のサイトと考え，正規のサイトが使用している難読化を良性の難読化として扱う．また，D3M データセットおよび，Malwr.com から取得したファイルを悪性の難読化として扱う．

正規のサイトおよび悪性のサイトを調査した結果として，以下の特徴を得ることができた．良性の難読化では主にランダム難読化とデータ難読化が利用されており，エンコード難読化がほとんど利用されていないことが確認できた．また，コードの可読性を維持するための改行やインデントといったものが利用されていないサイトが多く見られた．悪性の難読化ではランダム難読化，データ難読化，エンコード難読化が利用されており，多くの場合でエンコード難読化によって攻撃コードの本体が隠されていると考えられる．一部ではエンコード難読化後にデータ難読化が行われているのが確認できた．

	Legitimate	Malicious
Normal 0-100	4,859	16,566
Normal 100-400	2,328	600
Random & Data	5,672	275
Encode	10	58

表 4: 分類結果

結果から静的解析による難読化の分類を行った．分類の手法として，1 行あたりの文字列の長さとカンマやセミコロンといった区切り文字を利用した．また，分類の結果から動的に html を生成する JavaScript がエンコード難読化に混ざったため，動的に html を生成するものを取り除くために div やコロン，http といった文字列を利用して分類した．図 6 は分類手法である．この結果 (表 4) からエンコード難読化の抽出は可能であり，今回の結果では誤検知が 10 件，見落としが 8 件存在した．また，条件を変更することで 4 件の見落としについては改善可能であり，残りの 4 件については組み合わせ型の難読化が利用されていた．

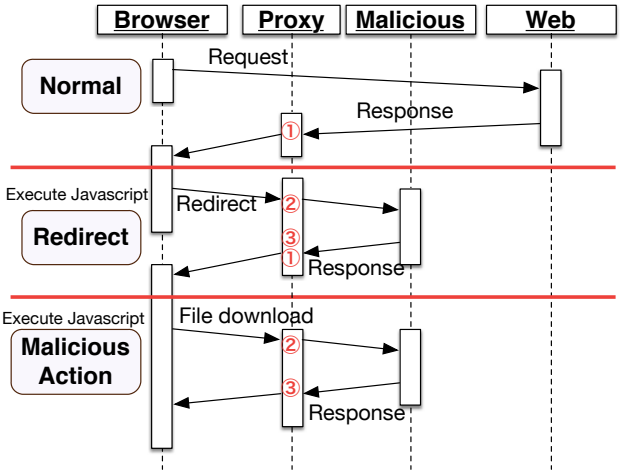

図 7: 提案システム

4.2 提案手法

ドライブバイダウンロード攻撃の検知手順として，ユーザとウェブサイトの間に検知用のプロキシを設置し，プロキシ上でウェブサイトから得られる html や js といったスクリプトが書かれたファイルと DNS の問い合わせ情報を取得し．悪意のある通信の検知に利用する．悪意のあるサイトで利用されているエンコード難読化では，リダイレクト先となるドメイン情報を含めた全ての情報を別の文字列へ置き換えている．そのため，ウェブサイトから得られるファイルでは通信先のドメイン情報が難読化によって確認することができない．ドメイン情報が確認できないことを利用して，DNS の問い合わせで発生する通信からドメイン情報を取得し，取得したファイルの中にドメイン情報が記述されているかで通信を判断する．

図 7 で示す提案する検知の手順は以下となる．

1. 難読化ファイルの取得：ユーザがウェブサイトに問い合わせを行い，レスポンスで得られる難読化された html や js といったファイルをプロキシ上で取得する．
2. ドメイン情報の取得：ユーザがウェブサイトから取得した難読化されたファイルのスクリプトがウェブブラウザ上で実行され，難読化の解除が行われる．これにより，難読化されていたドメイン情報の DNS 問い合わせが発生する．
3. ドメインの検索：スクリプトによって発生した DNS の問い合わせをプロキシで取得し，事前に取得しておいた難読化されたファイル内に問い合わせのあったドメインが記述されているか確認を行う．記述されていない場合は悪意のある通信として検知し，通信を停止させることで，ユーザの端末へのマルウェア感染を防ぐことが可能である．

なお，提案手法については文献 [4] において発表されている．現在は提案手法を Google Chrome のプラグインとして実装中である．

5 おわりに

本年度の SWAN Working Group の活動は，悪性ウェブサイト対策技術についての研究を行った．なかでも，その代表的な類型であるフィッシングサイトならびに難読化マルウェア配布サイトに焦点を当てて研究を行った．来年度も幅広い種類の悪性ウェブサイトについて，多面的な解析を行っていく．

参考文献

[1] Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi, *"An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites,"*, *Australian Journal of Intelligent Information Processing Systems*, Vol. 10, No. 2, pp.54-63, November 2008.

[2] Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi, *"HumanBoost: Utilization of Users' Past Trust Decision for Identifying Fraudulent Websites,"*, *Journal of Intelligent Learning Systems and Applications*, Vol. 2, No. 4, pp.190-199, December 2010.

[3] Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi, *"EyeBit: Eye-Tracking Approach for Enforcing Phishing Prevention Habits,"*, *In Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, September 2014.

[4] 藤原 寛高, ブラン グレゴリー, 櫨山 寛章, 門林 雄基, **難読化の特徴を利用したドライブバイダウンロード攻撃検知についての検討**, 電子情報通信学会技術報告, Vol. 114, No. 340, ICSS2014-60, pp. 55–60, 2014 年 11 月.