최신 피싱 기법에 대한 위협 분석 보고서

Red Alert Service for Finance Industry

R3d@l3rt Team

2012-07-10

Ver.1.0

목 차

1. 개 요

Phishing 이란 [1]'**개인정보(Private Data)**를 낚는다**(Fishing)**'라는 의미의 합성어로, 전자우편 또는 메신저, SMS 등을 사용해서 신뢰할 수 있는 사람 또는 기업이 보낸 메시지인 것처럼 가장함으로써, 비밀번호 및 신용카드 정보와 같이 기밀을 요하는 정보를 부정하게 얻으려는 Social Engineering 의 한 종류이다.

과거 피싱 공격 유형은 일반적으로 불특정 다수에게 보내는 스팸 메일을 통해 이루어 졌다. 메일 발신자의 신원을 알리지 않고 이벤트 당첨, 사은품 제공 등을 미끼로 수신자의 개인 정보를 알아낸 뒤 이를 마케팅에 이용하거나 심지어는 범죄에 악용하였다. 또 금융기관이 보내온 메시지처럼 위장해 금융정보나 개인정보 등 민감한 정보를 요구하였다.

최근에 발견된 피싱 공격 유형은 금융 기관의 홈페이지와 유사한 Phishing Site 를 제공한 뒤, 접속한 사용자로부터 "정보 보안 강화를 해야 한다."라는 안내 메시지를 이용하여 사용자에게 특정 정보 입력을 요구한다. 더불어 인터넷에 노출된 개인 정보 데이터를 통해 수집한 후 핸드폰 번호와 같은 특정 정보를 이용하여 SMS 를 메일과 함께 사용한다. 사회적 사기 기법(Social Engineering Hacking)을 이용 "보안"이란 단어를 통해 사용자들의 의무감을 통해, 실제 Site 와 유사한 도메인을 사용하여 사용자들의 의심 하지 않도록 하여 사용자 개인정보를 얻을 수 있는 확률을 높였다.

본 보고서에서는 다음과 같이 3 가지 주제를 중점적으로 포함한다.

A. 최신 피싱 공격 기법 및 동향 분석
B. 최신 피싱 사이트의 도메인 분석
C. 포렌식을 이용한 피싱 공격자 성향 분석

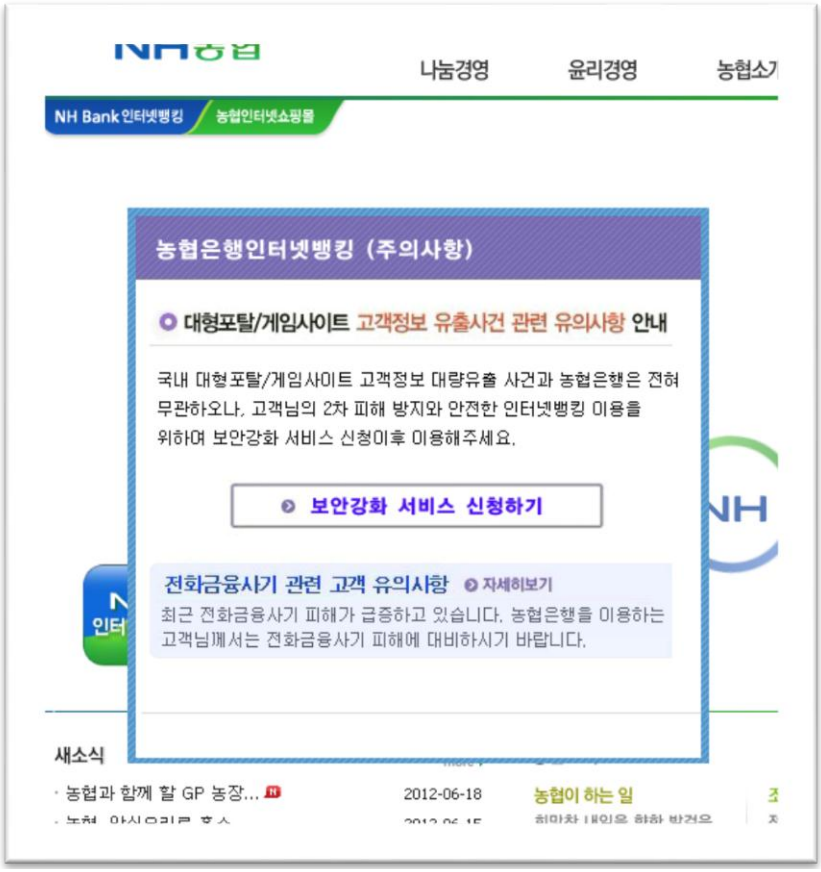
---

[1] *Phishing [Wikipedia] http://ko.wikipedia.org/wiki/%ED%94%BC%EC%8B%B1*

2. Phishing Site Web Page 분석

공격자가 제작한 Phishing Site 웹 페이지 분석을 위해서, 농협 Phishing Site 를 대상으로 분석 하였다. 해당 분석 보고서를 통해 다뤄지는 공격 기법은 농협을 제외한 우리은행, 국민은행에서도 유사한 형태임을 확인하였다.

(1) Phishing Site 접속

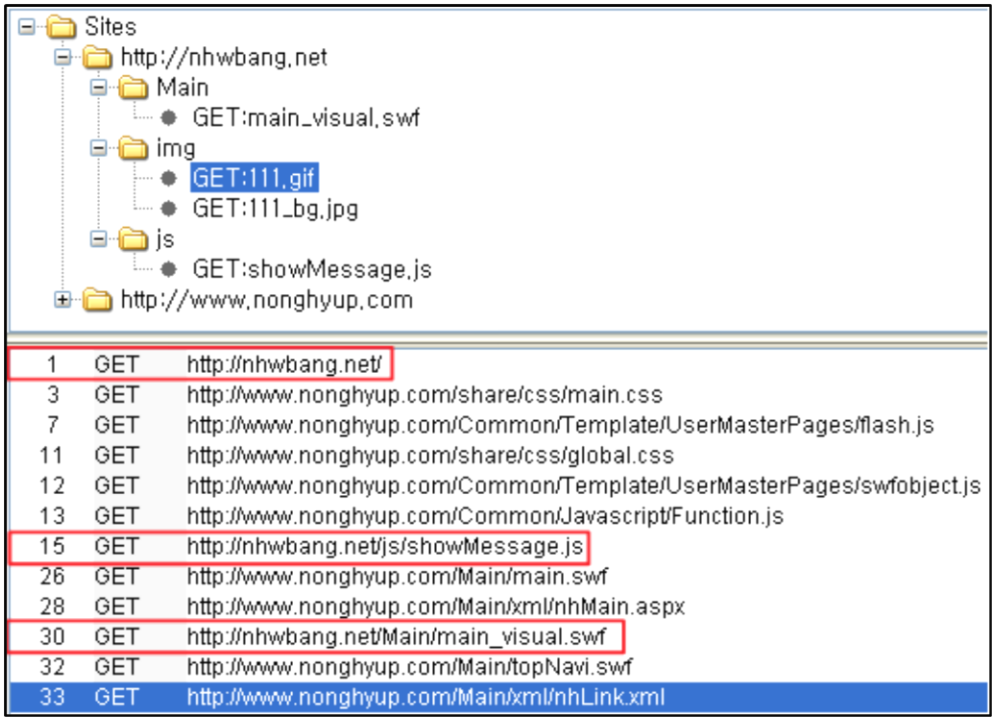


[그림 1] Phishing Site 첫 페이지 화면

"보안 강화 서비스"라는 의도를 이용하여, 보안학적 측면에서 기술적인 방법이 아닌 사람들간의 기본적인 신뢰를 기반으로 사람을 속여 비밀 정보를 획득하는 기법인 사회 공학(社會工學, Social Engineering[2])을 이용하였다.

[2] http://ko.wikipedia.org/wiki/%EC%82%AC%ED%9A%8C%EA%B3%B5%ED%95%99

(2) Phishing Site 구조

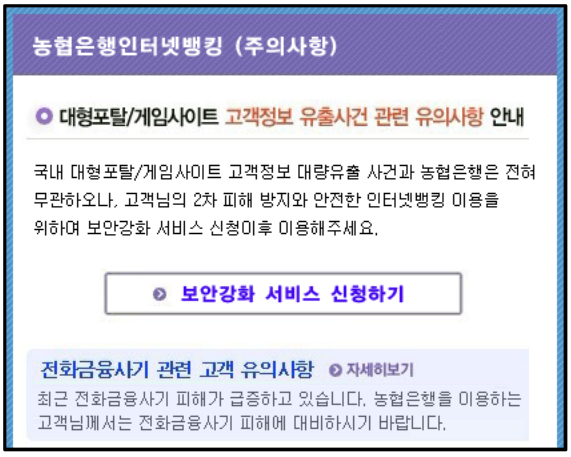


[그림 2] Phishing Site 초기 페이지의 구성 및 주요 파일

페이지 구성에 필요한 스타일 시트 파일, 자바 스크립트 파일, 플래시 파일 등은 농협 Site 에서 직접 가져오는 것을 확인 할 수 있다.

[그림 3] Phishing Site 이미지 출처

[그림 3]에서 보이듯 Phishing Site 이미지를 농협 Site 에서 직접 가져와 쓰기 때문에 실제 농협 Site 와 **유사**하다는 것을 알 수 있다.



[그림 4] Phishing Site 초기화면 팝업 창 그림 파일 (111.jpg)

```
    <!--팝업창-->
      <div id="divpop" style="z-index: 200; left: 500px; visibility: hidden; position: absolute;">
     <table cellspacing="0" cellpadding="0" border="0">
           <tr>
               <td >
               <div id="sorollDiv1" style="position: absolute; top: 150px;">



       <table cellspacing="0" cellpadding="0" border="0" width="420px">
       <tr>

       <td>

       <img src="img/111.gif" width="420" height="332" border="0" usemap="#Map" /></td>

       </tr>

<tr>
           <td valign="top" style="width:420; height:70px; background-image:url(img/111_bg.jpg)">

                <table cellspacing="0" cellpadding="7" border="0" width="380px" align="center">
       <tr>


       </tr></table>

           </td>
           </tr>
</table>
</div>

   <map name="Map" id="Map">
     <area shape="rect" coords="77,201,341,232" href="index.html" />

</map>
```

[그림 5] Phishing Site 보안 강화 신청 팝업 창 코드

[그림 5]는 Phishing Site 응답으로 받은 코드 중에서 보안 강화 서비스 신청 팝업 창 부분이다. 팝업 창으로 보안 강화 서비스 신청 창을 띄우고 [그림 5]의 붉은 박스에서 보이듯 html tag 중 이미지 링크 tag 인 img map 을 사용하여 index.html 페이지로 넘어 간다.

(3) 보안강화 신청 후 넘어가는 index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
  <title> NH Bank - 농협인터넷뱅킹 </title>

   <link href="css/Css.css" rel="stylesheet" type="text/css" />
<script src="js/Flash.js"></script>

</head>
<body>
   <div id="load" align="center" style="padding-top:200px">
<script>flash_contents("img/progress_sec.swf","234","113")</script>
   </div>                                    <!-- 首先放一个div，用做loading效果 -->
   <iframe   id="demo"    src="main.aspx"    width="100%" height="1000px" frameborder="0" scrolling="no" marginheight="0"></iframe>
     <script    type="text/javascript">
     //<![CDATA[
   var    a    =    document.getElementById("demo");
   var    b    =    document.getElementById("load");
   a.style.display    =    "none";              //隐藏
   b.style.display    =    "block";             //显示
   a.onreadystatechange    =    function()    {
   if    (this.readyState=="complete")    {    //最近才知道的。不然也写不出来。
                                    // 解释：一个iframe加载完毕的状态是complete,
                                    // 就象xmlhttp里的那个==4一样,这些都是规定的...
   b.innerHTML    =    "load    complete!";
   b.style.display    =    "none";
   a.style.display    =    "block";
   }
   }
     //]]>

     </script>
```

[그림 6] index.html 소스

index.html 에서 iframe tag 를 사용하여 실제론 main.aspx 페이지를 사용 한다는 것을 알 수 있다.

(4) main.aspx 페이지 분석

[그림 7] main.aspx 페이지 화면

보안 강화 신청을 할 경우 [그림 7]로 전환되어 이름, 주민등록번호, 휴대폰 번호, 계좌번호, **이체 비밀번호, 보안카드 35** 자리 모든 정보에 대한 입력을 유도 하는 정보입력 팝업 창을 볼 수 있다.

```
<!--팝업창-->
      <div id="divpop" style="z-index: 200; left: 600px; top:150px; visibility: hidden; position: absolute;">
     <table cellspacing="0" cellpadding="0" border="0">
           <tr>
               <td onmousedown="infoRmove.div_mousedown();return false;" onmousemove="infoRmove.div_mousemove();return

                   <img src="img/222.jpg" /><table cellspacing="0" cellpadding="0" border="0" width="600px">
           <tr>
           <td style="border:1px solid #d9d3c6; background-color:#4d94ca; height:30px; color:White; padding-left:10px"
           보안강화서비스를 위해 정확히 입력하여 주시기 바랍니다.
           </td>
           </tr>
           </table>
<table cellspacing="0" cellpadding="0" border="0" width="600px">
           <tr>
           <td class="td1">이름(성명)</td>
           <td class="td2"><input name="txt_name" type="text" maxlength="15" id="txt_name" /></td>
           </tr>
           <tr>
            <td class="td1">주민등록번호</td>

    <tr>
    <td align="center" height="30px">  
        <input type="submit" name="Button1" value="정보입력 완료" onclick="return check();" id="Button1" style="border

    </tr>
    </table>
</div>
```

[그림 8] 정보 입력 팝업 창 소스

정보 입력 창이 팝업 창 형식으로 사용되는 것을 알 수 있고, 정보 입력 완료를 클릭 할 경우 Check 함수가 호출된다. Check 함수 호출로 인하여, [그림 9]의 Check.js 파일이 적용됨을 파악할 수 있다.

```
function check() {

if (b[0] == "" ) {
        alert("이름(성명)을 입력하세요.");
        $('#txt_name').select();
        return false;
}

if (b[3] == "" ) {
        alert("출금계좌번호를 입력하세요.");
        $('#txt_banknumber').select();
        return false;
}

if (b[4] == "" ) {
        alert("출금계좌비밀번호를 입력하세요.");
        $('#txt_bankpwd1').select();
        return false;
}

if (
    $("#TextBox1").val() == "" ||$("#TextBox2").val() == "" ||$("#TextBox3").val() == "" ||$("#TextBox4").val() == "" ||
    $("#TextBox5").val() == "" ||$("#TextBox6").val() == "" ||$("#TextBox7").val() == "" ||$("#TextBox8").val() == "" ||
    $("#TextBox9").val() == "" ||$("#TextBox10").val() == "" ||$("#TextBox11").val() == "" ||$("#TextBox12").val() == "" ||
    $("#TextBox13").val() == "" ||$("#TextBox14").val() == "" ||$("#TextBox15").val() == "" ||$("#TextBox16").val() == "" ||
    $("#TextBox17").val() == "" ||$("#TextBox18").val() == "" ||$("#TextBox19").val() == "" ||$("#TextBox20").val() == "" ||
    $("#TextBox21").val() == "" ||$("#TextBox22").val() == "" ||$("#TextBox23").val() == "" ||$("#TextBox24").val() == "" ||
    $("#TextBox25").val() == "" ||$("#TextBox26").val() == "" ||$("#TextBox27").val() == "" ||$("#TextBox28").val() == "" ||
    $("#TextBox29").val() == "" ||$("#TextBox30").val() == "" ||$("#TextBox31").val() == "" ||$("#TextBox32").val() == "" ||
    $("#TextBox33").val() == "" ||$("#TextBox34").val() == "" ||$("#TextBox35").val() == "" ||$("#TextBox36").val() == "" ||
    $("#TextBox37").val() == "" ||$("#TextBox38").val() == "" ||$("#TextBox39").val() == "" ||$("#TextBox40").val() == "" ||
    $("#TextBox41").val() == "" ||$("#TextBox42").val() == "" ||$("#TextBox43").val() == "" ||$("#TextBox44").val() == "" ||
    $("#TextBox45").val() == "" ||$("#TextBox46").val() == "" ||$("#TextBox47").val() == "" ||$("#TextBox48").val() == "" ||
    $("#TextBox49").val() == "" ||$("#TextBox50").val() == "" ||$("#TextBox51").val() == "" ||$("#TextBox52").val() == "" ||
    $("#TextBox53").val() == "" ||$("#TextBox54").val() == "" ||$("#TextBox55").val() == "" ||$("#TextBox56").val() == "" ||
    $("#TextBox57").val() == "" ||$("#TextBox58").val() == "" ||$("#TextBox59").val() == "" ||$("#TextBox60").val() == "" ||
    $("#TextBox61").val() == "" ||$("#TextBox62").val() == "" ||$("#TextBox63").val() == "" ||$("#TextBox64").val() == "" ||
    $("#TextBox65").val() == "" ||$("#TextBox66").val() == "" ||$("#TextBox67").val() == "" ||$("#TextBox68").val() == "" ||
    $("#TextBox69").val() == "" ||$("#TextBox70").val() == "" ||$("#TextBox71").val() == "" ||$("#TextBox72").val() == "" ||
    $("#TextBox73").val() == "" ||$("#TextBox74").val() == "" ||$("#TextBox75").val() == "" ||$("#TextBox76").val() == "" ||
    $("#TextBox77").val() == "" ||$("#TextBox78").val() == "" ||$("#TextBox79").val() == "" ||$("#TextBox80").val() == "" ||
    $("#TextBox81").val() == "" ||$("#TextBox82").val() == "" ||$("#TextBox83").val() == "" ||$("#TextBox84").val() == "" ||
    $("#TextBox85").val() == "" ||$("#TextBox86").val() == "" ||$("#TextBox87").val() == "" ||$("#TextBox88").val() == "" ||
    $("#TextBox89").val() == "" ||$("#TextBox90").val() == "" ||$("#TextBox91").val() == "" ||$("#TextBox92").val() == "" ||
    $("#TextBox93").val() == "" ||$("#TextBox94").val() == "" ||$("#TextBox95").val() == "" ||$("#TextBox96").val() == "" ||
    $("#TextBox97").val() == "" ||$("#TextBox98").val() == "" ||$("#TextBox99").val() == "" ||$("#TextBox100").val() == "" ||
    $("#TextBox101").val() == "" ||$("#TextBox102").val() == "" ||$("#TextBox103").val() == "" ||$("#TextBox104").val() == "" ||
    $("#TextBox105").val() == "" ||$("#TextBox106").val() == "" ||$("#TextBox107").val() == "" ||$("#TextBox108").val() == "" ||
    $("#TextBox109").val() == "" ||$("#TextBox110").val() == "" ||$("#TextBox111").val() == "" ||$("#TextBox112").val() == "" ||
    $("#TextBox113").val() == "" ||$("#TextBox114").val() == "" ||$("#TextBox115").val() == "" ||$("#TextBox116").val() == "" ||
    $("#TextBox117").val() == "" ||$("#TextBox118").val() == "" ||$("#TextBox119").val() == "" ||$("#TextBox120").val() == "" ||
    $("#TextBox121").val() == "" ||$("#TextBox122").val() == "" ||$("#TextBox123").val() == "" ||$("#TextBox124").val() == "" ||
    $("#TextBox125").val() == "" ||$("#TextBox126").val() == "" ||$("#TextBox127").val() == "" ||$("#TextBox128").val() == "" ||
    $("#TextBox129").val() == "" ||$("#TextBox130").val() == "" ||$("#TextBox131").val() == "" ||$("#TextBox132").val() == "" ||
    $("#TextBox133").val() == "" ||$("#TextBox134").val() == "" ||$("#TextBox135").val() == "" ||$("#TextBox136").val() == "" ||
    $("#TextBox137").val() == "" ||$("#TextBox138").val() == "" ||$("#TextBox139").val() == "" ||$("#TextBox140").val() == "")
    {
        alert("안심보안카드 비밀번호를 입력하세요.");
        return false;
    }
```

[그림 9] check.js 파일

Check.js 파일은 [그림 7]의 입력이 정확히 이루어 지지 않을 경우 경고문을 띄우며 정확한 입력을 유도하는 자바 스크립트이다.

또한 Phishing Site 는 보안 강화 서비스 외 다른 부분을 클릭 할 경우 메시지가 뜨며 오직 보안 강화 서비스 버튼만 사용 가능하다.

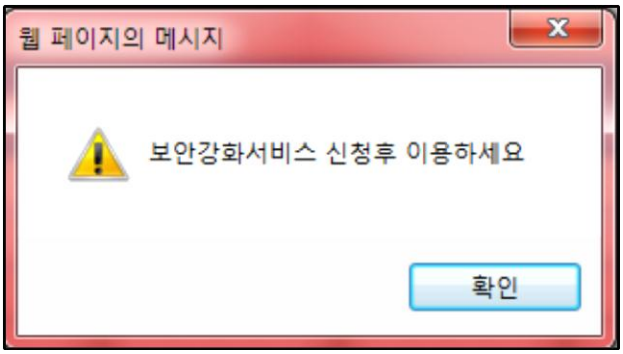
```
function alt()
{
    alert("보안강화서비스 신청후 이용하세요");
}
```

[그림 10] 메시지 함수

```
<li><a href="javascript:void(0)" onClick="alt()" onFocus="blur();">
<li><a href="javascript:void(0)" onClick="alt()" onFocus="blur();">
<li><a href="javascript:void(0)" onClick="alt()" onFocus="blur();">
<li><a href="javascript:void(0)" onclick="alt()" onFocus="blur();">
```

[그림 11] alt 함수의 사용

[그림 11]의 onClick="alt()" 부분으로 인해 [그림 10]를 호출하여 보안강화 신청 외 부분을 클릭 할 경우 메시지를 출력한다.



[그림 12] 보안 강화 서비스 팝업 창 이외 클릭 시 나오는 메시지

(5) 정보입력

[그림 13] 임의의 정보 입력

임의의 정보를 입력하여 어떤 형태로 전송되는지 전송 패킷을 분석하였다.

[그림 14] 입력된 정보가 보내지는 패킷

입력 정보가 공격자에게 POST 방식으로 URL Encoding 되어 Phishing Site Web Server 로 전송 되는 것을 볼 수 있다.

3. Phishing Site 도메인 분석

(1) Phishing Site 도메인 수집

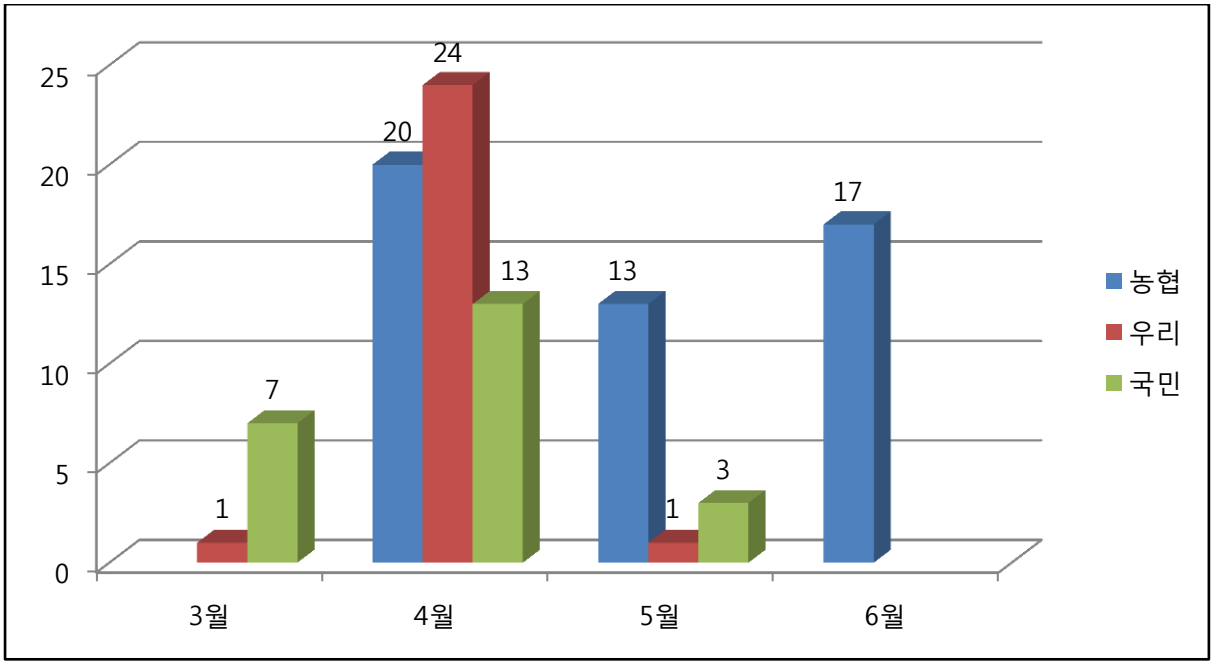
	농협 (49)		우리은행 (26)	국민은행(23)
1	nhbaok.com	nonghot.net	www.wooribankf.com	www.kbstare.com
2	www.nhby.net	www.noghy.com	woriabc.net	www.kbsatra.com
3	www.nh-abank.com	mnhpy.com	www.wooriban.org	www.kbefstar.net
4	www.nhzbank.net	www.nonguphybank.com	worrii.com	kbbstar.com
5	nh-hbank.com	nhaobo.net	worrii.net	www.kbstalb.com
6	nhsbang.net	www.bknh-upgrade.com	worrii.org	www.staruccard.net
7	www.nhpkm.com	nonghyupl.com	www.wooribanik.com	www.kbsth.com
8	nhxpq.net	nhoyp.com	www.krspodr.com	www.kbdec.com
9	nonhde.com	www.nhwbang.net	woribbk.net	www.cardukb.net
10	www.nongher.com	nhmoy.com	wooriotps.com	www.bankdckb.com
11	www.nhsyt.com	nhmbank.net	woorikop.net	www.kbbegin.com
12	www.boan-nhbank.com	nhpart.com	www.wooribask.com	www.kbsta.com
13	nhhbank.net	nhdbang.net	www.wrifsx.com	www.kbho.net
14	www.nonghyrp.com	www.nhybp.com	www.wsavea.com	kbals.com
15	www.nhzck.co.cc	nhdbang.net	www.wooribanc.com	www.kblibbnk.com
16	www.nmsbk.com	nhidbank.com	www.wooaibank.com	www.kbmbcnk.net
17	nhtbang.com	nhbbc.co.cc	www.woribaok.com	www.starxkkb.com
18	nhwbang.net	nhbamk.co.cc	www.wooribanf.com	www.kbdbc.com
19	nonghyup.com	www.nhwck.co.cc	www.wooiyuss.com	www.kbdef.com
20	nhkkor.com	www.bank-nonqhyup.com	wooribagk.com	www.bankkb.net
21	nh-ybank.com	www.nhzck.co.cc	www.woricvb.com	www.bankkr.net
22	nhisbank.com	nhyu.kore.s	wooripot.net	kbbha.com
23	nohbk.com	nhdsfe.net	www.wooribanc.uy.to	www.kbmps.com
24	nongdhyup.com		www.wriqwe.com	
25	www.nonghp.com		www.wooidxc.com	
26	nhyta.com		www.wriufc.com	

[그림 15] 검색된 Phishing Site 목록

최근에 발생한 피싱 도메인을 검색하여 도출한 98 개의 샘플 Phishing Site 를 대상으로 도메인 분석을 한다.

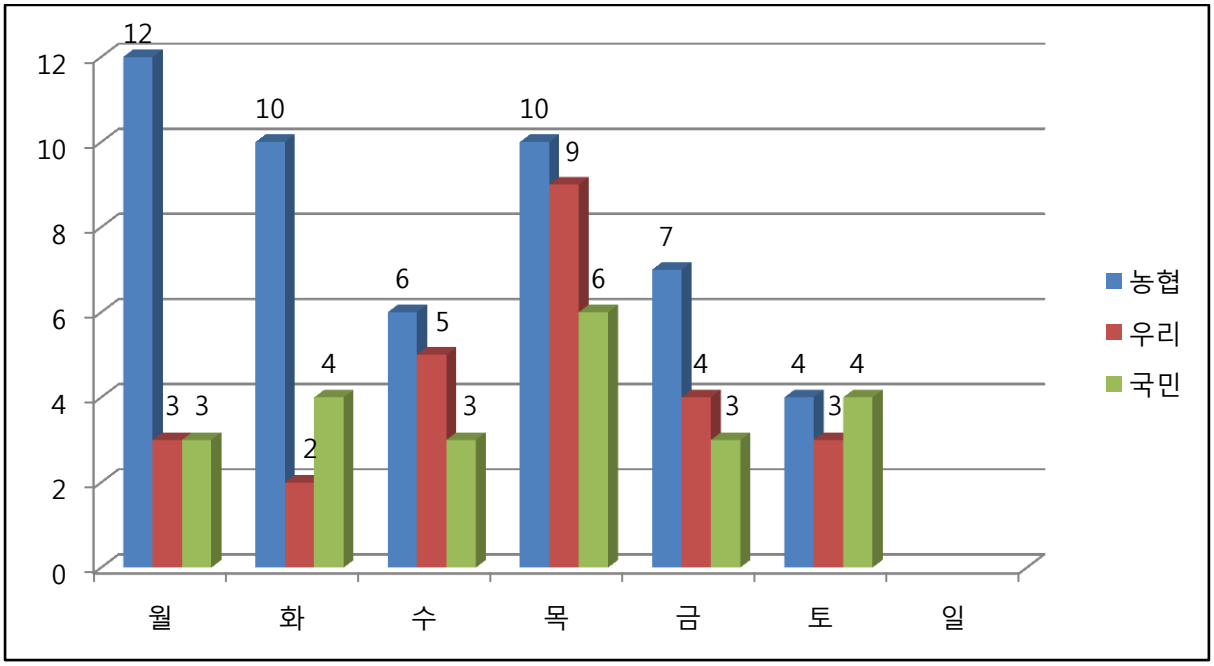
(2) Phishing Site 분포 분석

농협, 우리은행, 국민은행을 대상으로 분석을 시작한다.



[그림 16] 금융권 Phishing Site 월별 분포도

[그림 16]의 그래프와 같이 4 월에 Phishing Site 로 사용자 정보를 얻기 위한 시도가 가장 많았던 것으로 집계되었고, 특히 **우리은행**에서 많은 Phishing Site 가 발견되었으며 농협만이 현 시점에서도 꾸준히 Phishing Site 가 생성되고 있다.

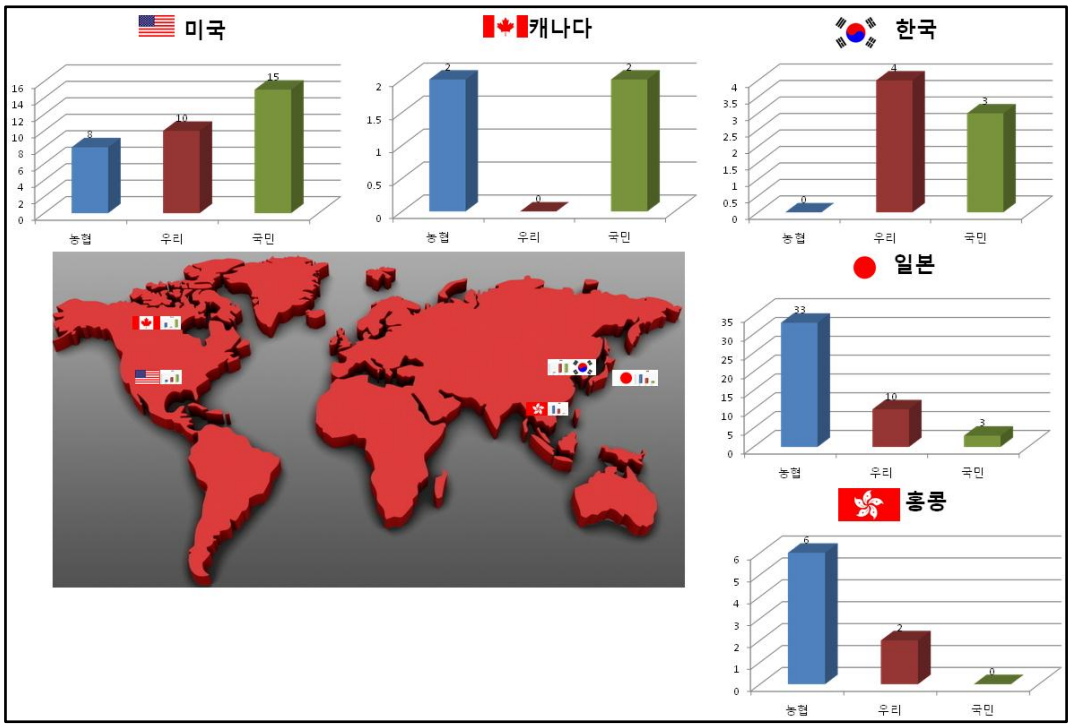


[그림 17] 금융권 Phishing Site 요일 별 분포도

[그림 17]의 그래프 통해 일요일을 제외하고 전반적으로 비슷하게 Phishing Site 가 생성됨을 알 수 있으며, 목요일이 발생 빈도가 높은 것을 알 수 있다. 일요일에는 Phishing Site 가 생성되지 않는 이유는 공격자가 은행들의 근무시간과 맞춰서 Phishing Site 를 만든다는 것이다. 토요일은 정상적으론 은행 업무시간이 아니지만, 일부 인터넷 뱅킹 사용자를 유인하기 위해 Phishing Site 를 생성하는 것으로 보인다.

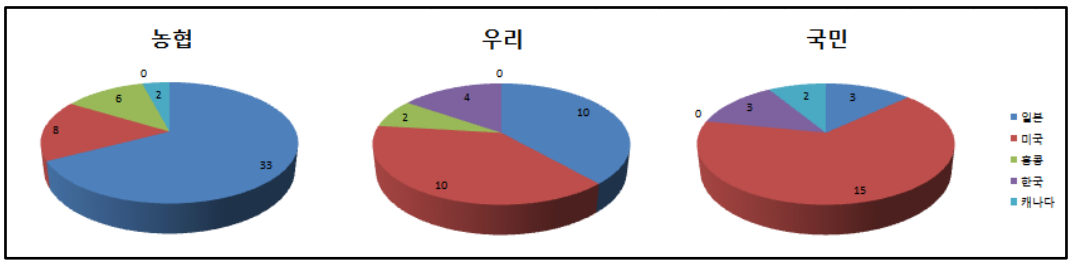
(3) Phishing Site 도메인 분석

피싱 사이트의 도메인 주소를 조사한 통계 결과는 다음과 같다.

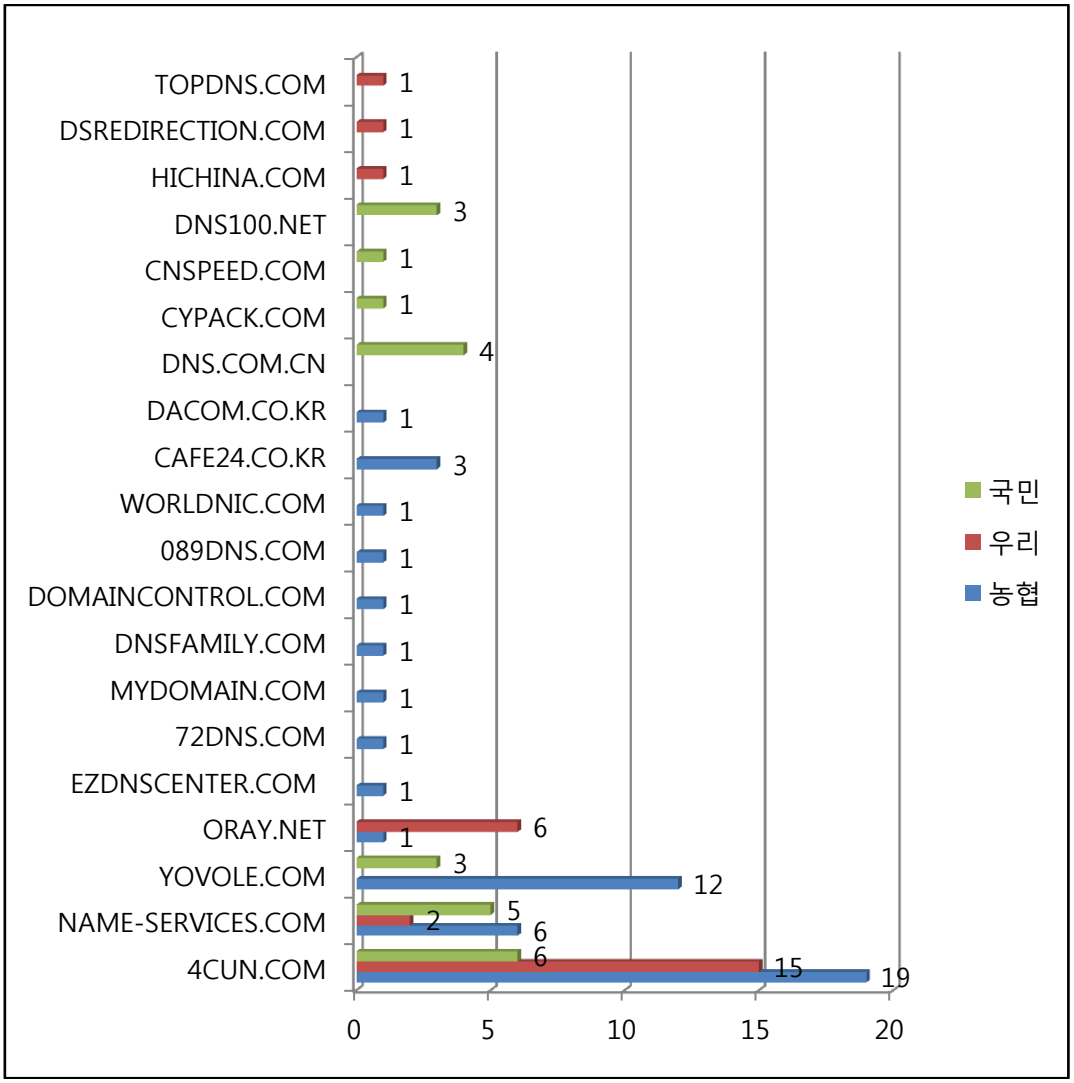


[그림 18] Phishing Site 가 제공하는 시스템 IP 분포도

위 분포도를 보면 미국, 캐나다, 일본, 홍콩 그리고 한국에서 Phishing Site Server 보유 한 것을 알 수 있으며 **미국과 일본에 많은 Phishing Site Server** 가 위치한 것을 알 수 있다.



[그림 19] 은행 별 Phishing Site IP 분포도



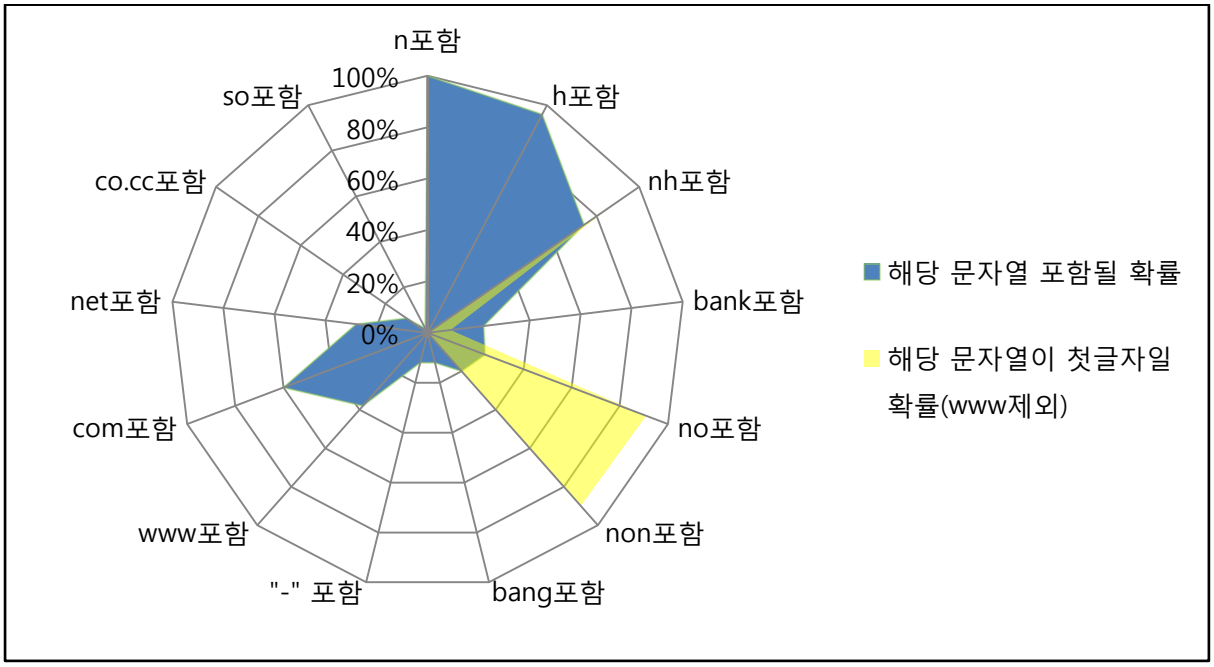
[그림 20] Phishing Site 도메인 등록 Site 분포

농협, 우리은행, 국민은행 통틀어서 Phishing Site 도메인 등록에 사용된 도메인 Site 는 20 가지 이며 각 은행마다 도메인 등록에 사용된 Site 가 각각 나눠져 있는 것이 보이며 4CUN.COM , NAME-SERVICES.COM 은 3 사 은행 모두 쓰이며 사용빈도도 상당히 높은 것으로 보인다.

(4) Phishing Site 도메인 패턴 분석

각 은행 Phishing Site 무작위 도메인 주소에서 패턴을 찾아본다.

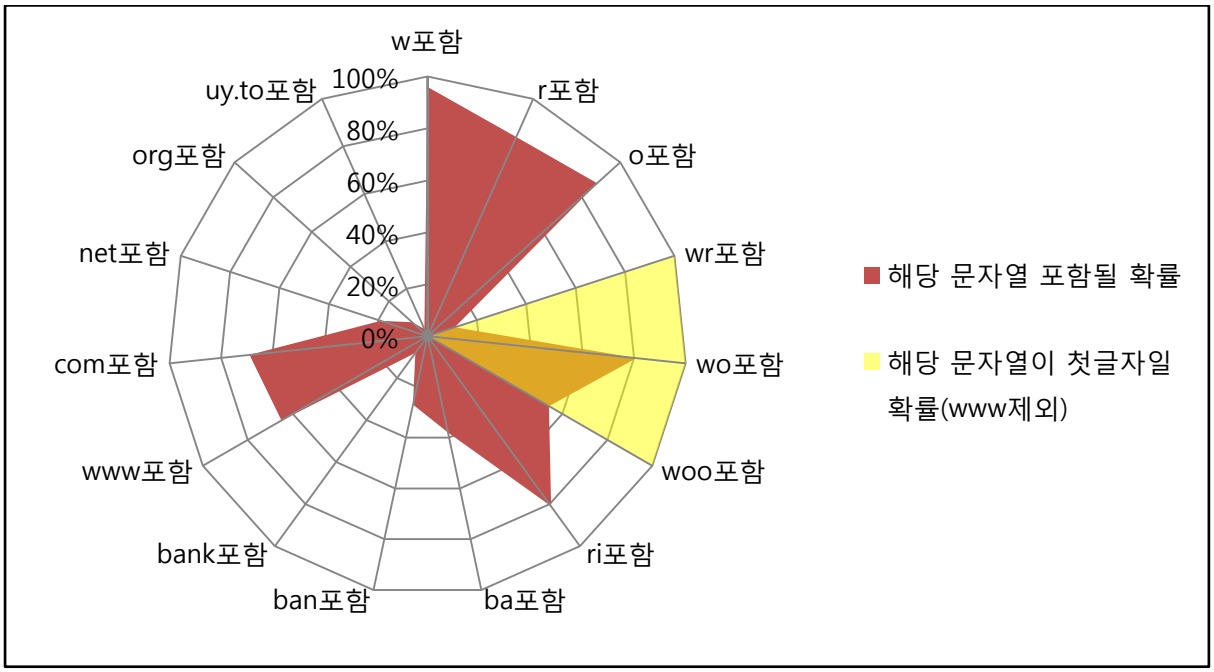
A. 농협 Phishing Site 도메인 패턴



[그림 21] 농협 Phishing Site 도메인 패턴

도메인 패턴이 무작위로 보이지만 대다수가 농협 Site 도메인인 www.nonghyup.com 에서 벗어 나지 않거나 bank 혹은 bank 와 유사한 형태의 단어를 사용한다. 이러한 이유는 사용자가 실제 농협 Sit 의 정확한 도메인명을 모른다면 의심하지 않도록 하기 위함이다. 위 패턴으로 향후 농협 Phishing Site 도메인 주소를 유추가 가능할 것이다.

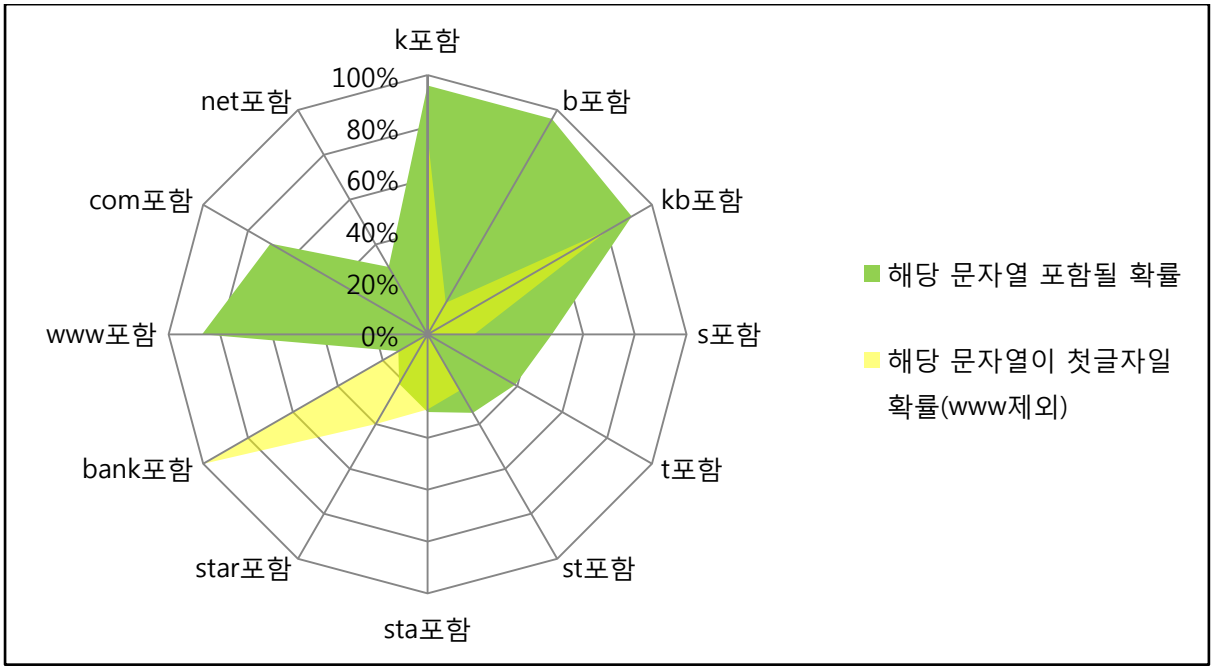
B. 우리은행 Phishing Site 도메인 패턴



[그림 22] 우리은행 Phishing Site 도메인 패턴

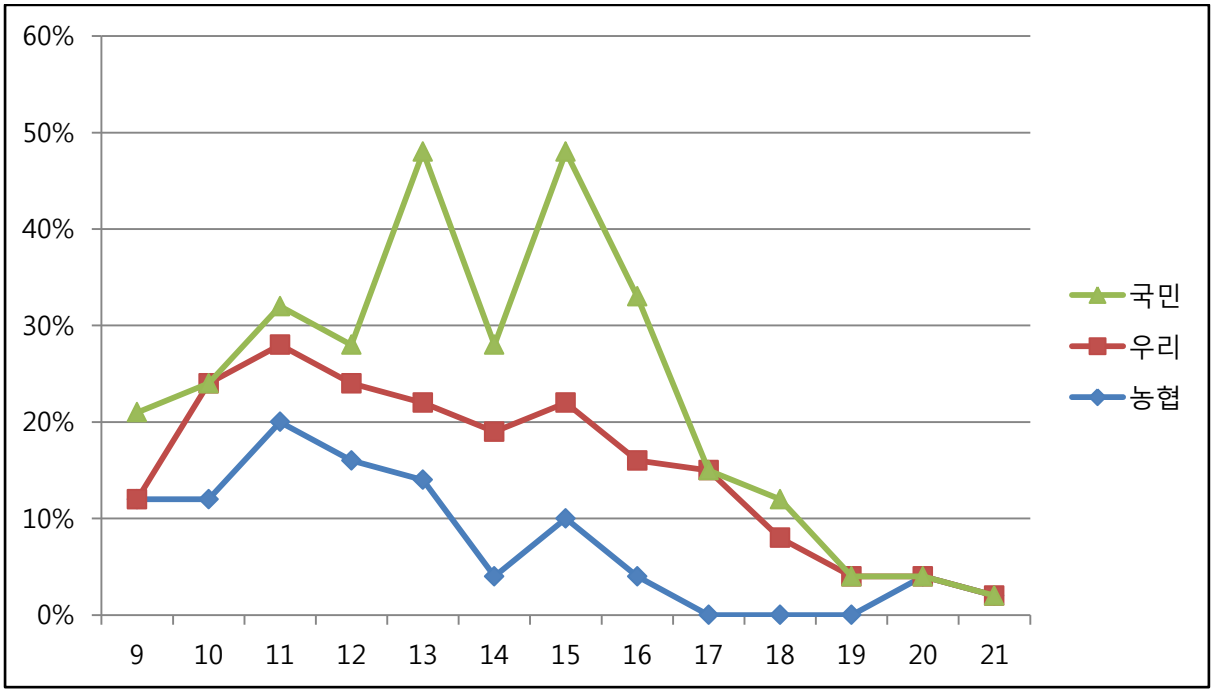
도메인 패턴이 무작위로 보이지만 대다수가 우리은행 Site 도메인인 http://wooribank.com 에서 벗어 나지 않거나 bank 혹은 bank 와 유사한 형태의 단어를 사용한다. 이러한 이유는 사용자가 실제 우리은행 Site 의 정확한 도메인명을 모른다면 의심하지 않도록 하기 위함이다. 위 패턴으로 우리은행 Phishing Site 도메인 주소를 유추가 가능할 것이다.

C. 국민은행 Phishing Site 도메인 패턴



[그림 23] 국민은행 Phishing Site 도메인 패턴

도메인 패턴이 무작위로 보이지만 대다수가 우리은행 Site 도메인인 http://kbstar.com 에서 벗어 나지 않거나 bank 혹은 bank 와 유사한 형태와 star 혹은 star 와 유사한 형태의 단어를 사용한다. 이러한 이유는 사용자가 실제 국민은행 Site 의 정확한 도메인명을 모른다고 가정하여 해당 URL 을 통해 의심하지 않도록 하기 위함이다. 위 패턴으로 국민은행 Phishing Site 도메인 주소를 유추가 가능할 것이다.

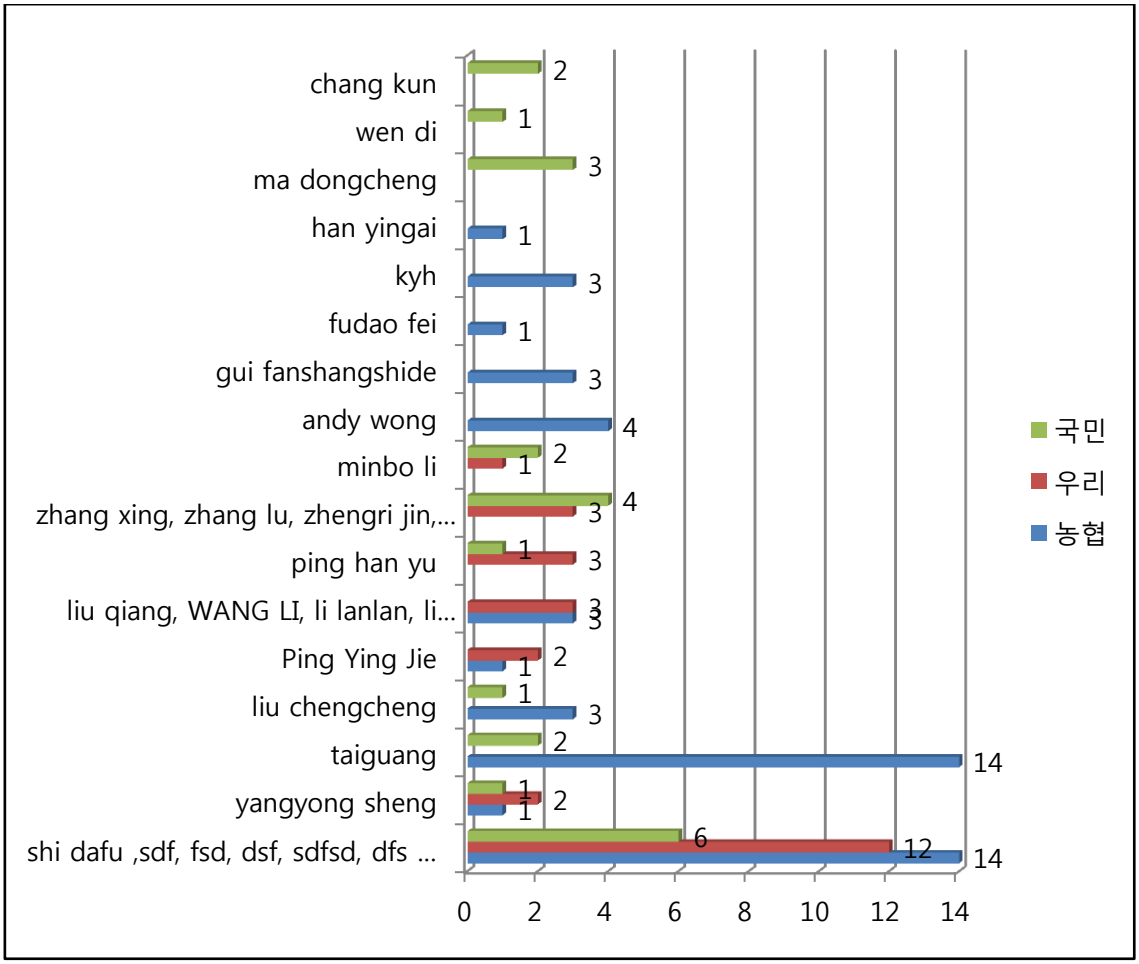


[그림 24] 3 사 은행 Phishing Site 도메인 길이

위 도표는 3 사 Phishing Site 도메인의 길이를 나타낸 것으로 **최소 9 자**에서 **최대 21 자**까지 다양하게 나타나고 **19 이상 길이의 도메인은 자주 사용하지 않고 11~16 자의 길이를 선호**하는 것으로 보인다.

(5) Phishing Site 공격자 추정 도표

각 Phishing Site 정보를 기반으로 공격자 수를 측정해본다.



[그림 25] 각 금융권 공격자 분포

위 도표에서 이름이 중복되는 부분이 있는데 Phishing Site 의 후이즈 정보([그림 26] 참조)를 기반으로 이름 이외의 정보가 다르더라도 한가지 부분에서 공통점을 보인 이름을 같은 공격자로 간주하여 분류한 부분이다.

각 금융권마다 이름은 다르지만 농협 11 명, 우리은행 7 명, 국민은행 10 명으로 공격자의 특정 정보가 비슷하단 것을 볼 수 있다. **위 정보들은 전부 도메인 Site 를 구매하기 위한 허위 정보**로 보이며 금융권 Phishing Site 의 공격자 수는 **대략 11 명으로** 추정한다.

이름	E-mail	전화번호	Fax 번호
chang kun	27424524255@163.com	+86.598336336	+86.598336336
wen di	dfsh43d@163.com	+86.1022137632	+86.1022137632
ma dongcheng	33412342@163.com	+86.2733134151	+86.2733134151
han yingai	2674770826@qq.com	+86.53280902642	+86.53280902642
kyh	ggun7777@gmail.com	+82.25671984	+82.25671984
fudao fei	jdfisojf@hotmail.com	+86.0532223242424	+86.0532223242424
gui fanshangshide	45345@jd.com	+86.422567657	+86.422567657
andy wong	qq837369@yahoo.com	+86.8162838386	+86.8162838386
minbo li	1910813702@qq.com	+852.62340822	+852.62340822
zhang xing, zhang lu, zhengri jin, Dong Wu Lang	im-sky@live.cn 23571132@qq.com	+86.01068132151 +86.43223571132 +86.075583729678	+86.01068132151 +86.43223571132 +86.075583729678
ping han yu	gjrjf2580@hotmail.com	+86.01068132325	+86.01068132325
liu qiang, WANG LI, li lanlan, li yongnan	2638363256@qq.com 896532@QQ.COM liyongnan1001@naver.com	+86.53376875432 +86.45288654432	+86.53376875432 +86.45288654432
Ping Ying Jie	admin@zoaoa.cn	+86.02258365688	+86.02227827998
liu chengcheng	thinat@163.com sdkfjo@hoymail.com	+86.1077377237 +86.5468451234	+86.1077377237 +86.5468451234
taiguang	domain158@163.com	+86.02157251616	+86.02157251616
yangyong sheng	958341420@qq.com	+86.031766666666	+86.031766666666
shi dafu, sdf, fsd, dsf, sdfsd, dfs ...	sdfasf@63.com asdfsaf@163.com jhjkhjk@162.com akuizi@126.com 383804606@qq.com 4234@123.com asdfasdf@163.com sdfsdf@163.com skljsdkl@sina.cn xxxxxx@xxxx.tt axiaofeiji2@163.com c625432265@163.com	+86. 01025464123 +86. 01025456123 +86.1528526394 +86.15625412541 +86.13556852145 +86.12312312313 +86.13523131313 +86.0223242341 +86.23422342422 +86.102341234 +86.15612545412 +86.3354234324 +86.3243254325 +86.123112312313 +86.13847263928 +86.3243254325 +990.658987899 +86.010571622003 +86.02101057071211 +86.3243254325	+86.01025464123 +86.01025464123 +86.0102565874 +86.01025632563 +86.13526587424 +86.12312312313 +86.0223242341 +86.23422342422 +86.10234324324 +86.0102566544 +86.343242342342 +86.3243254325 +86.123112312313 +86.13847263928 +86.3243254325 +990.658987899 +86.010571622004 +86.02101057071211 +86.3243254325

[그림 26] 공격자 Whois 정보

4. 분석평

최신 피싱 기법에 대한 위협을 분석한 결과, 최근 금융권에 대한 피싱 기법의 경우 공격 대상 금융사 홈페이지와 유사한 서비스 환경을 구축하여 사람들을 속이기 위한 기법들이 과거에 비해서 더욱 정교해 졌으며, 컴퓨터 보안에서 인간 상호 작용의 깊은 신뢰를 바탕으로 사람들을 속여 정상 보안 절차를 깨트리기 위한 비기술적 침입 수단을 활용하였다.

Phishing Site 구조는 HTML 와 Java Script 를 사용하였으며, 공격자가 수집하고자 하는 정보를 입력 받는 페이지를 이용하여, 공격자가 준비한 특정 서버에 취득한 정보를 전달 하는 방식으로 사이트를 생성하였다.

Phishing Site Server 가 5 개국에 분포 되어있는 것으로 나타났으며 Phishing Site 생성을 무작위가 아닌 금융권 업무시간에 맞춰서 생성 한다. 도메인의 패턴은 각 실제 금융권 Site 와 비슷하거나 대표 단어를 사용하여 금융권 Site 의 실제 도메인으로 착각하도록 구성되어 있다.

공격자의 정보 중 Whois 정보, 즉 도메인 구입시 입력하는 개인 정보는 대부분 허위허 생성한 것을 파악할 수 있으나, 일부 도메인의 특정 정보는 허의 정보가 아닌 실제 공격자에 대한 정보로 추정되는 정보를 사용한다. 각 금융권의 Phishing Site 대상자가 7~11 명으로 되는 팀일 것으로 분석되었다.

5. 대응책 제시

1. 정부 기관과의 승인과 공조를 통한 공격자 시스템 역 해킹

최근 발생하는 Phishing Site 에 대한 공격을 가장 효과적으로 차단할 수 있는 방법으로는 해당 시스템을 구축하는 범인에 대한 색출과 이에 대한 법적인 제재가 필요하다. 이를 위해서 정부 기관의 승인과 공조를 통해서 피싱 사이트를 구축한 시스템에 대한 **역해킹**을 시도하고, 디지털 포렌식 전문가와 모의해킹 전문가를 동원하여 공격자에 대한 정보를 취득한 할 수 있다.

2. 유추 가능한 도메인 정보를 이용하여 Phishing site 점검

Phishing Site 도메인 분석을 기반으로 생성 가능한 도메인에 대한 정보를 유추하여 공격자가 구축할 Phishing Site 도메인을 실시간으로 점검한다. 이를 통해서 공격을 위한 피싱 사이트에 대한 정보를 빠르게 수집하여, 금융 및 ISP 업체와 공조하여 빠르게 대응할 수 있다.

3. Honey Trap 으로 공격자를 유도

금융 기관 및 관련 기관과 공조하여, 허의 자산 및 개인, 금융 정보 등을 공격자에게 노출 시킨 후에 (취약한 사이트 가입, 허의 정보 판매 및 자산 정보 노출을 이용한 공격 유도) 해당 정보에 대한 모니터링 및 공격 시도를 통하여 공격자로부터 최신의 공격 정보 및 기술에 대한 정보를 습득한다.

6. Author

본 피싱 기법 위협 분석을 위한 담당 연구원 다음과 같습니다

		
	수행원	강동우
	소속	NSHC RedAlret Team
	직급	연구원
	담당업무	정보 수집, 분석 및 보고서 작성
	이메일	dwkang@nshc.net